THE UNIVERSITY OF ALBERTA

[illegible]

by

© [illegible]

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES AND RESEARCH
IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE
OF [illegible]

[illegible]

EDMONTON, ALBERTA

[illegible]

THE UNIVERSITY OF ALBERTA

PATTERN RECOGNITION WITH OPTICAL TRANSFORMS AND SURFACE FITTING

by

© Paul G. Sorenson

A THESIS
SUBMITTED TO THE FACULTY OF GRADUATE STUDIES
IN PARTIAL FULFILMENT OF THE REQUIREMENT FOR THE DEGREE
OF MASTER OF SCIENCE

DEPARTMENT OF COMPUTING SCIENCE
EDMONTON, ALBERTA
Fall , 1969

UNIVERSITY OF ALBERTA

FACULTY OF GRADUATE STUDIES

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies for acceptance, a thesis entitled PATTERN RECOGNITION WITH OPTICAL TRANSFORMS AND SURFACE FITTING submitted by Paul Gordon Sorenson in partial fulfilment of the requirements for the degree of Master of Science.

ABSTRACT

This thesis is divided into two distinct phases; the design of a pattern preprocessor and the development of a trainable pattern classifier. The preprocessor consists of a combined system. A coherent light source (laser light) is focused on an object (e.g. a letter) and then is passed through a concave lens. The power spectrum of the light at the transform plane is combined with the light pattern of the object and is fed into the pattern classifier. A surface fitting model is used to categorize the characteristic surfaces formed by the preprocessor. A possible hardware configuration of the model is outlined in some detail, and a simulation study is completed on over eighteen hundred hand-printed characters.

ACKNOWLEDGEMENTS

I express my appreciation to Professor Gordon H. Syms for his advice, criticism and guidance during the preparation of this thesis; to Professor J.R. Sampson and Professor I.N. Chen for their helpful suggestions to improve the final draft; and to Sandy Paul for time spent in typing the thesis.

I wish to thank the Department of Computing Science for the financial assistance and facilities which have made this project possible.

TABLE OF CONTENTS

Page

LIST OF FIGURES

LIST OF TABLES

CHAPTER I

THESIS INTRODUCTION

1.1 Introduction

As digital computers increase in speed and complexity, they become more and more predominant in the automated society of today. At present, almost all facets of industry and commerce are aided by these electronic devices. With the greater demand for production of and versatility in computing machines, no bounds can be placed on their input requirements. Neither is there a limit on the amount of information that machines output for humans to digest and base their decisions on. Good or bad, the emphasis is now on automation and automation demands better and faster communication between man and machines. There is a demand for machines that can be commanded by voice, machines that diagnose diseases and assist in operations, and machines that read hand printing and handwriting. Slowly improvements in communication are being developed in these and other areas where there exists a man-machine interface.

This thesis is devoted to the problem of recognizing hand-printed characters. To date, few hand-printed character recognition machines have been designed which yield acceptable results (80% or better). The machines which do achieve good

results are extremely slow or work on a small character set. Few of these machines have been produced commercially. Therefore, one of the main pursuits of this research is the development of a fast, accurate yet unsophisticated, hand-printed character recognition machine - a device which might be considered for commercial services.

The main difficulty which arises in the development of a Hand Printing Recognition System (HPRS), lies in the characters, themselves. Every person has a unique style of printing. Some styles become so totally personalized that other readers are challenged by what is written. Fortunately, humans have the ability to extract information from context. This ability makes their recognition rate far superior to a machine's which as yet do not possess this capability. For example, in the sentence

The c-t r-n home

the letters a in the words cat and ran are only identifiable from context.

If the ability to extract contextual information is so important why then should it not be included in HPRS? Ideally it should - but as yet no fast, accurate system exists which identifies well-formed characters on which the contextual information for ill-formed characters depend. In

this thesis, we shall "attempt to walk before running", thus restricting our research to be area of noncontextual recognition.

In order to generalize the character environment so that HPRS is capable of handling an almost infinite number of different fonts, some scheme of extracting invariant properties within a class (eg. all hand-printed letter A's) must be developed. The first stage of the HPRS, the preprocessor, is responsible for extracting these invariant properties. In the thesis a new type of preprocessor the Fourier transform preprocessor (PREP), is introduced to determine if it can find the common properties for each class of characters. The second stage of HPRS, the categorizer, generalizes the character environment from the preprocessor information obtained during the training period. Once the training period is completed, the categorizer should be able to correctly classify patterns from the same character environment as the training set. From some extensions to the theory of Piecewise Linear (PWL) classification, a new categorizer or learning machine (LEM) is developed and tested. The categorizer is a surface fitting model and employs prototype matrices to represent categories in pattern space. Thus it can be stated the immediate interest and purpose in this research effort is the exploration of a new, unique hand printing recognition system (HPRS). It should be kept in mind that character recognition belongs to the more general

field of pattern recognition. Therefore, any developments in this thesis are also appliciable to a wide range of pattern recognition problems.

1.2 Outline by Chapters

For the purpose of orientation and to justify the development of HPRS, some present approaches to the problems of preprocessing and the problems of training and classification are surveyed in Chapter II. Chapter III is a presentation of the theoretical concepts behind the HPRS model. Included is a summary of the theory of the optical transform and some heuristical developments in the theory of the PWL classifier. In Chapter IV a proposed hardware implementation and the computer simulation of HPRS are outlined. As well, a machine definition for HPRS is conceived with the aid of formal, set theory notation. The performance of the simulated HPRS model is discussed in Chapter V. The effects of different machine parameters are also demonstrated. In the final chapter, Chapter VI, some conclusions are drawn from the exploration of the HPRS model. Suggestions are made for the improvement of the present model and the direction of further development in the field of character recognition.

CHAPTER II

A SURVEY OF THE FIELD OF CHARACTER RECOGNITION

A character recognition system consists of two stages, the preprocessing stage and the categorizing stage. A discussion of three different preprocessing techniques is included in this chapter. In Sec. 2.2 three of the most popular methods of solving nonlinearly separable pattern recognition problems are outlined. Both Sec. 2.1 (preprocessing techniques) and Sec. 2.2 (categorizing methods) conclude with a brief analysis and a short description of the HPRS preprocessor and categorizer to be developed in this thesis.

2.1 Preprocessing

The term "preprocessing" has acquired a variety of meanings. In this thesis preprocessing shall be referred to as any activity involving the transforming of data from the input pattern to a pattern acceptable to a classifier. During this transformation, the data may undergo alterations such that the input to the classifier may no longer be a representative picture of the character. For example, some preprocessors extract features from the pattern and output a vector of feature weights which typify the pattern. To

account for such changes in the measurement domain,* a TRANSFORMATION MEASURE shall be defined as the number of transform mappings of the input information as performed by a preprocessor. A transform of an nxn matrix of points (M) (representing the outline of the pattern (P)) into a vector of characteristic weights (V), can be demonstrated by the following mapping,

$$t_1 : M_p \rightarrow V_p$$

where t_1 is a transformation mapping of measure 1.

A transformation mapping of measure zero implies that there is no change in the measurement domain. The template type of preprocessor is an example of a preprocessor of transformation measure zero. In the next section a detailed description is given of the template preprocessor. A preprocessor which extracts features from the optical transform (frequency domain) of a character would be an example of a preprocessor with a transformation mapping of measure two. This concept shall be mentioned further with reference to each of the three preprocessor techniques

---

*A change in the measurement domain implies a change in what is measured. That is, a value of 1 in (an element of) the matrix, M_p, (above) may signify the presence of the character at that position; whereas, a value of 1 in the vector, V_p, may describe the presence of a straight line in P.

to be surveyed in this chapter.

In character recognition, noise may be described as unwanted distortions in the input pattern. These distortions are from two main sources. One source (TYPE ONE NOISE) is inherent in the formation of the character whether by typewriter ribbon, pencil lead, or the human's inability to write properly. A second form of distortion (TYPE TWO NOISE) is due to the transmission of data by light waves or electric current. Because this research is in the simulation stage, the problems of type two noise cannot be competently discussed. However, the different type one noises shall be outlined.

There are basically five type one noises. These are: figure rotation, figure translation, figure scaling, variable line thickness and smudging or faintness. When describing the template, feature and holographic preprocessing techniques, we shall discuss each one's effectiveness in handling these noise problems.

2.1.1 The Template Approach

In template preprocessing no attempt is made to extract any pattern characteristics other than the form of the pattern itself. Consequently, in using the template approach there is no change in the measurement domain-implying that this type of preprocessing has a transformation measure of zero.

Basically, in template matching the area correlation between the unknown pattern and each of the typical patterns from the identifiable set is measured. This can be accomplished by two different methods: the raster method, or the masking method. In the raster method light is radiated from the pattern onto a retina of light sensitive photocells*. The photocells are either binary (the output is 1 or 0 depending whether the amount of light striking the cell is above or below a preset threshold) or multivalued (the output is set to a grey scale value which is proportional to the amplitude of the light striking the cell). The output from the photocell raster is sent to the categorizer. The categorizer has available stored representations of typical patterns. It employs these patterns as well as some form of decision logic to render a classification.

In the masking method the light from the pattern is focused on a series of photographic masks which represent typical patterns. The light that passes through the mask falls onto a retina of photocells of the form described previously. In this instance the output from the preprocessor

*Magnetic character recognition machines also use the raster method. In this case when the magnetic ink of the character comes in contact with a raster element which conducts electricity, a current is produced indicating the ON condition.

is a direct measure of the area correlation between the photographic mask and the unknown pattern.

It is easy to visualize that this simple preprocessing technique could result in pattern misrepresentation. Rotated and translated patterns, unless registered properly, will yield small area correlations with a prototype pattern. Undoubtedly this will result in misclassification by the categorizer. Characters printed in different scales present a great deal of difficulty to the template preprocessor. To accommodate this difficulty a prescan would have to be employed to detect the scaled characters so that they may be enlarged or reduced to the appropriate size. The template approach is extremely sensitive to changes in the thickness of the lines used to represent the pattern. Too thick of a line may enhance the correlation between an input pattern and a typical pattern of a wrong category. Too thin of a line may fail to produce an adequate degree of correlation between the input pattern and a typical pattern of the correct category. This could result in rejection. Patterns subject to heavy noise because of corrected printing mistakes or smudges are usually misclassified. Faintness results in a rejected character since the character's outline is not fully represented.

Rosenblatt's "Perceptron" [Minsky (1963)] is the classical example of a pattern recognition machine using a template preprocessor. Most of today's commercial machine-printed recognition systems employ the template type of preprocessing. Some examples are the Control Data 915 Page Reader, the Multiple Head MICR IBM #1419, and the Control Data Ft. Monmouth Multi-font Reader, AN/FST-6 [Holt (1968)]. Most of these machines use a prescan technique to detect and help eliminate the type one noise.

2.1.2 The Features Approach

In general, a feature extracting preprocessor is a preprocessor of transformation measure one. It attempts to transform the physical shape of the pattern into a list of features which can adequately represent the pattern. The major difference between this approach and the template approach is that a pattern is considered as a collection of basic parts rather than a whole entity. Recognition is achieved by noting the set of basic features of an unknown pattern and comparing that set with the typifying sets of learned patterns. There exists two kinds of feature preprocessors: those that use special hardware and those that employ computer software (programs) to extract features. We shall describe an example of each.

The Farrington Credit Card Reader is a commercial

character recognition machine (numbers only) which employs hardwired feature detecters. The form of the character is transferred by light onto a raster of photocells. Vertical registration is accomplished by a mechanical scanner which scans in cooperation with triggering timing circuits. The electrical output signals from this system are connected to sets of flip-flops each representing a feature such as LEFT VERTICAL BAR, TOP HORIZONTAL BAR, etc. The output from these flip-flops are passed to the categorizer. The feature detectors are hardwired and do not enjoy enough flexiability to preprocess characters that are rotated or have variable line thicknesses. Generally the hardwired-type of feature preprocessor is inadequate for hand-printed text, but function quite well for type-written material.

The next system to be introduced is designed by John H. Munson to recognize hand-printed text [Munson (1968)]* Munson's machine uses computer programs to extract features from hand-printed Fortran coding sheets ready for keypunching. The hand-printed characters are scanned by a vidicon television camera operated under the control of SDS 910 computer. Each

---

*The Munson system shall be presented in detail because the data used in the extensive testing of the HPRS model originates from the multiple-coder file. This is a file that Munson collected and used in the testing of his own model.

document is mounted in a concave cylindrical holder so that, as the camera scans across the document, the viewing distance and hence the image scale remains constant. The camera generates a standard closed-circuit television waveform, which is quantized to two levels (black/white) by a Schmidt trigger. A sample raster contains 120x120 points.

A document is illuminated by four flood lights mounted around the TV camera. A colored filter is placed over the camera lens to suppress the colored coding-sheet guideline. The field of view is chosen so that a single character image is usually slightly less than 24 points high and about 15 points wide. The computer begins the scanning by reading in a 120x120 picture containing several character images. A scanning routine then proceeds approximately horizontally across the picture finding and isolating character images. As each character is isolated, it is placed in a standard 24x24 raster format. No corrections for magnification or rotation are applied.

Munson's feature detection method is embodied in two computer programs, PREP and TOPO. The PREP program performs edge detection on the 24x24 quantized image through the use of mask pairs or templates. Each mask pair consists of two adjacent 2x8 rectangles of points. One of the masks is given a positive weight the other a negative. A threshold is set such that if the positive mask encounters six more figure

points than the negative one, the binary response of the mask pair is ON. To provide a limited degree of translation invariance the response of five such adjacent mask pairs of the same orientation are OR-ed together to give a single binary component of the output feature vector. For each pattern a 84-bit feature vector is formed.

The TOPO preprocessor is a collection of computer routines assembled to extract topological and geometric features from the character image. These features describe the presence, size, location and orientation of enclosures, concavities, and stroke tips in the character. The feature weights from the TOPO program are multivalued (0-100). The magnitude of the weight is proportional to the extent in which the character possesses that particular feature. The outputs from TOPO and PREP are fed into a pattern classifier.

From the above description it can be seen that Munson's system may lack the speed of operation but is far more flexible and sophisticated than the Farrington hardware model. Here, there is an effort to eliminate the problems of rotation and translation. Since all of Munson's hand-printed text is written with standard HB-type pencil, an attempt is made to avoid a variety of line thicknesses. Faintness and smudging are noise problems Munson ignores. Unless the pattern is extensively degraded, the feature detector remains unaffected by this type of noise.

Two classic pattern recognition systems which employ the feature processing technique are Selfridge and Neisser's [1963] "Pandemonuium", and Uhr and Vossler's [1963] matrix operator program. The IBM 1287, Standard Print Number Reader, CDC Watchbird Hand Print Number Reader and IBM 1975 Omni-Font Reader are three commercial machines which use scanners and line trackers to extract features from the character they wish to recognize [Holt (1968)].

2.1.3 The Holographic Approach

In the holographic preprocessing technique, the spatial frequencies of a pattern are extracted and compared to a filter mask. The light from this process is reconstructed (returned to the spatial domain) and used in the identification of the pattern. Because a pattern is distinguished by its spatial frequencies (or optical transform), a holographic preprocessor has a transformation measure of one.

The holographic technique is optical in nature. A coherent light source (laser light)* is directed at a transparent object (see Fig. <2.1.3.1>) forming a light pattern of the object.

----

*A "point" source of monochromatic light and a lens to properly collimate the light may be a poor substitute for a laser beam.

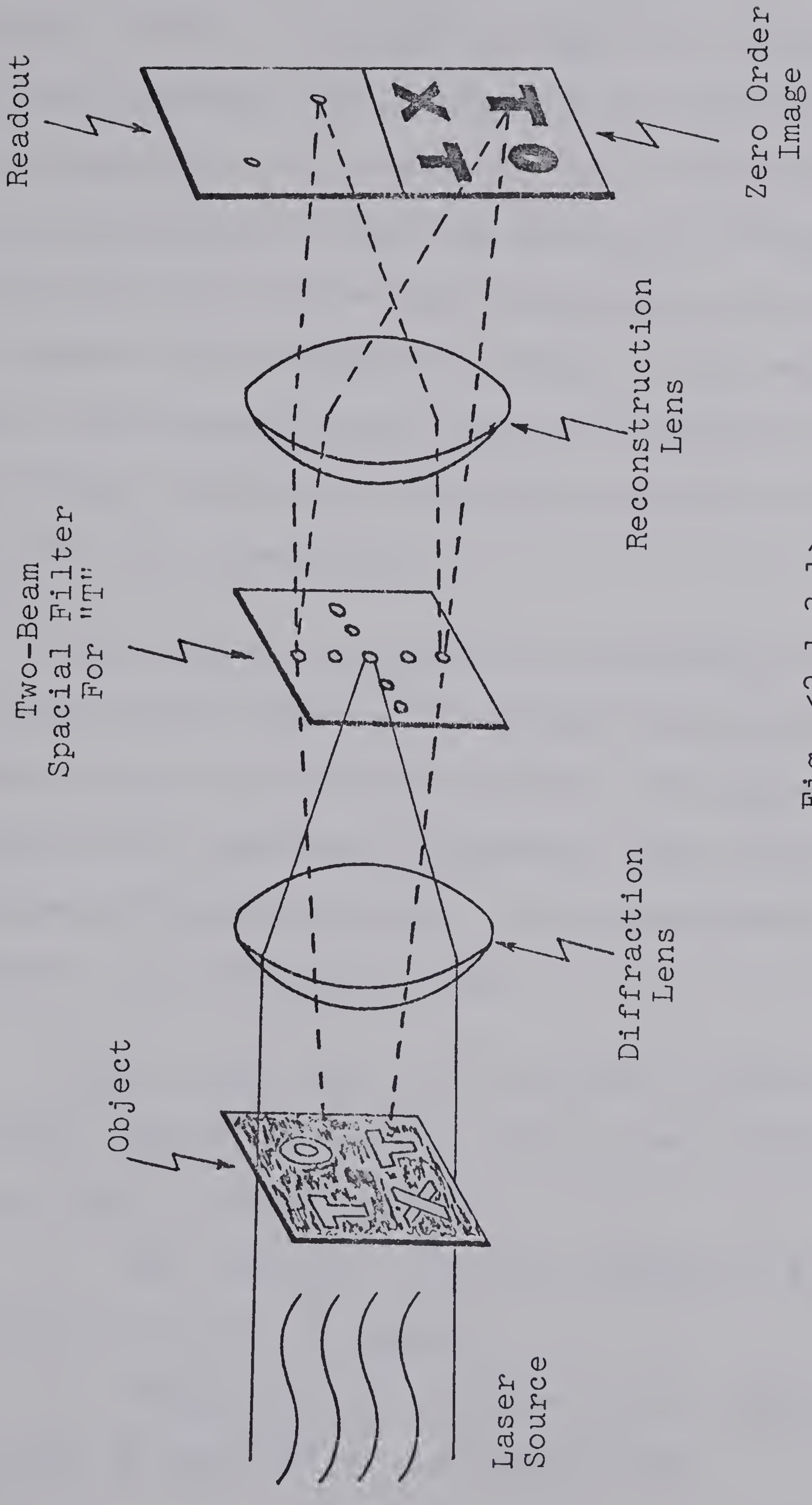


Fig. <2.1.3.1>

Holographic Recognition System

This pattern is passed through a spherical lens which focuses a diffraction pattern on the transform (filter) plane. At the transform plane the diffraction pattern falls on a spatial frequency filter or holographic mask of a typical pattern. The filter contains both a point focus (representing the zero order frequencies of the object) and a diffraction image of the typical pattern. When the filter is illuminated by the diffraction image of the unknown pattern, it produces a bright light called the "recognition image" at the reconstruction plane. The reconstruction plane is scanned by a photomultiplier which relays information about the intensity of the recognition image onto the categorizer.

When making a hologram or holographic mask both the amplitude and the phase of the light which passes through the transform plane is produced on film. The phase information is captured by "beating" a reference light source against the modulated diffraction pattern. The interference pattern formed is used as the holographic mask.

Vander Lugt [1968] in his paper on character reading by optical spatial filtering, states some limited advantages of this type of preprocessing.

1. The character could be recognized even if it were tilted plus or minus 5 degrees.

2. Recognition is obtained by the character varying up to plus or minus 5% of its normal size.

3. Recognition is obtained with mutilated characters.

4. Spatial filters containing edge information are superior to those containing area information.

In general, the holographic preprocessing technique suffers from the same noise problems as the template method - simply because it uses a holographic mask in comparing the unknown pattern to the set of typical characters. There is one important difference, however; and that is in the holographic system the diffraction image produced is invariant to translation. No matter where the unknown pattern is placed in the x-y plane the diffraction pattern remains constant.

Of the three types of preprocessing discussed in this chapter, the holographic preprocessing technique is the newest. As a result, very little work has been completed and as yet no commercial machine uses this idea. The necessity of a good coherent light source and object transparency has left commercial interest laging. All this is expensive! Some of the research completed todate (Holeman [1968], Lesen, et al. [1968], Dickenson and Watrasiewicz [1968]) suggests that machine-printed codes are dealt with successfully. Variances in size, shape, rotation and line thickness of hand-print script may be too complex to be coped with adequately by this system. Nowhere in the literature could a discussion of this subject be found.

2.1.4 Concluding Remarks

It is conceded by almost all researchers in the field of pattern recognition that preprocessors of transformation measure zero (TEMPLATE APPROACH) are inadequate for handling all but type-printed or well formed hand-printed character sets. In the design of a hand-printing recognition system a preprocessor of transformation measure greater than zero should be selected. Even the hardware feature-type models (such as used by the Farrington machine), unless built with greater sophistication and flexability, can not properly process hand-printed characters. Munson's computer-aided feature descriptors do perform well enough to identify both multiple-coder (84% accuracy) and single-coder (94% accuracy) hand-printed files, but require a large amount of time to process one character. In the holographic approach, the optical preprocessing system is fast and invariant to translated characters. Unfortunately the holographic masking technique makes this method sensitive to slight rotations, size distortions and variable line thicknesses.

This thesis will experiment with a new preprocessing technique (the Fourier Transform technique) which is entirely an optical process. The preprocessor extracts the diffraction image of the pattern and passes this to the categorizer. This system is more flexible and less complex than the holographic

system for there is no use of a holographic mask and no need to reconstruct a recognition image. Chapter 4 will describe in greater detail the simulation and implementation of the Fourier Transform preprocessor. Part of Chapter 3 is devoted to the optical theory of the Fourier Transform and a description of its computer algorithm.

2.2 Training and Classification-the Categorizer

After some necessary transformations and alterations, the preprocessor presents a representation of the character to the second stage of a pattern recognizer-the categorizer. In general, there are two types of categorizers:

1. Those that classify only.

2. Those that learn from a set of typical training patterns so that later they are capable of classifying patterns.

The first type of categorizer is rather uninspiring. There is no training period because the machine designer chooses the categories a priori. The chosen categories are usually based on a set of idealized patterns. This practice is unsatisfactory, for often these idealized patterns do not represent patterns that arise from the working environment. This is especially true when considering a hand-printed character recognition system. The decision logic for such categorizers is usually by some absolute means (eg. the

greatest or least amount of light transmitted).

In this research interest lies with the second type of categorizer, sometimes called a _trainable classifier_. There are two different training methods, parametric and nonparametric, for trainable classifiers. In parametric training, the training set is presented "in mass" to the categorizer. The categorizer is designed to gather parameters (such as means, variances, etc.) about the entire training set. These parameters are then used in the formulation of a classification rule. In this chapter, as an example of a classification technique which uses parametric training, Sebestyen's generalized discriminant analysis will be discussed.

In nonparametric training, the training set is usually presented to the categorizer a pattern at a time. The categorizer decides whether the pattern is classified correctly or not. If a pattern is classified correctly, the next pattern is introduced. If a pattern is classified incorrectly, an adjustment is made to the categorizer so that if the pattern is presented in the next step, it is classified correctly. (This type of procedure is also termed adaptive.) How and to what extent the categorizer is adjusted depends on the

type of adaptive strategy it employs. In this section two of the most popular nonparametric adaptive algorithms - Hunt's CLS (Concept Learning System) and Nilsson's PWL (Piecewise Linear) discriminant technique - will be surveyed.

2.2.1 CLS (Concept Learning System)

The basic strategy of CLS, as described by Hunt, *et al.* [1966] in the book *Experiments in Induction*, is the *wholest* algorithm. The algorithm is simply stated.

> ....Take as the first trial hypothesis the set of all attributes and values present on the first positive instance. Call this the focus. Compare the focus with the set of attributes or values present on the next positive instance and set the focus equal to the intersection of the two. Repeat this on each positive instance, in succession, until all positive instances have been examined. The resulting focus can be used as a test to define the class membership.

Hunt divides the concept universe, U, into two parts, negative and positive instances. This is not applicable to a character recognition problem where the universe is divided into n parts (e.g. n=26 for the alphabet). That is, if the universe consists of the categories, R and not R, little is discovered about the identification of an unknown character if it is classified as belonging to not R. However, if each subuniverse* is considered as a

---

*A subuniverse is defined as an instance of a universe.

universe and the algorithm is applied to each of these new universes, U is further subdivided. By employing this technique recursively and stopping only when a subuniverse can be divided no further (implying category identification), the entire universe can be successfully dichotomized into n categories. For n categories this can be completed in at most n-1 steps.

At this point it would be helpful to present some of Hunt's notation which he used to designate a sample universe. Using this notation an algorithm for "calculating" a concept tree will be described. The concept tree is a description of a strategy to be employed on the pattern universe. The formulation of a particular strategy will be illustrated with an "all too-well designed" example.

Notation . . .

Let n be the number of attributes (features or characteristics) needed to describe all the categories in the universe uniquely. Let m be the number of different complete descriptions. An object may be described by a vector d, of dimension n, in which some of the elements may be the special symbols,?, (value unknown) or, #, (inapplicable). These later two cases may generally be disregarded. Given the description of an object, it should be possible to indicate the category to which the object belongs. This can be done by appending to the description vector, as the

n+1 symbol, a numerial to represent each category. Let the matrix A, used in describing the sample universe, be composed of m row description vectors. A^t is called the "full form" of the concept universe at time t. U^t, the universe description matrix, is an array of row vectors describing the universes (or subuniverses) calculated at time t.

Computing A Concept (Realizing Categories)

1) Let i=1,j=1

2) Examine each column of A^t for all members belonging to universe $U^t_{i.}$. If $U^t_{i.}$ has only one member then go to step 5.

3) Find the attribute that best divides the universe into two equal or almost equal subuniverses.

4) Place the description of these two subuniverses into rows $U^t_{j.}$,$U^t_{j+1.}$ of U^t.j indicates the row for the description of the universe to be created next. Store i and j in a special pointer array called POINT. Increment j by 2.

5) Increment i by one and check to see if i=j+1. If not go to step 1. If so, the computation is completed.

Once finished, the next pattern is introduced to see if CLS can correctly classify it. If CLS can, the next pattern is brought in. If it can not, the matrix A is

augmented with the pattern's description and a new concept is computed. This process is continued until a satisfactory accuracy is achieved. By using the information contained in the universe matrix, U, and the pointer array, POINT, a tree structure is defined which demonstrates the CLS strategy. As an example the matrix A^t and the computed concept tree are shown in Fig. <2.2.1.1>.

Limited Memory

Since a computer is a finite memory device and the wholist algorithm uses a large matrix, a large pattern sample will often fill the available core. In order to read in further patterns it is necessary to delete some of the patterns presently described in matrix A. A pattern is deleted on the basis of three characteristics: the success rate of the category to which the pattern belongs, the number of patterns representing the category to which the pattern belongs, and the time at which the pattern is introduced to CLS. Generally, the higher the success rate, the greater the number of representative patterns and the earlier a pattern is read in, the more likely it is chosen to be deleted.

Remarks

"A concept learning system is a device for creating a concept corresponding to some partition of a sample of objects which have been categorized by a pre-established

Fig. <2.2.1.1> Example A^t matrix

0: indicates the absence of that attribute

1: indicates the presence of that attribute

List of Attributes / Category Name	R	P	N	M	L	K	H	F	E	D	B
Full Hori. Line	0	0	0	0	1	0	1	1	1	0	0
Closure	1	1	0	0	0	0	0	0	0	1	1
Large Closure	0	0	0	0	0	0	0	0	0	1	0
Small Bottom Closure	0	0	0	0	0	0	0	0	0	0	1
More than one Vert. Line	0	0	1	1	0	0	1	0	0	1	0
Large Top Bay	0	0	1	0	0	1	1	0	0	0	0
Bottom Bay	1	0	1	1	0	1	1	0	0	0	0
More than one Horiz. Line	1	1	0	0	0	0	0	1	1	0	1
More than two Horiz. Lines	0	0	0	0	0	0	0	0	1	0	1
Category Number	11	10	9	8	7	6	5	4	3	2	1

Concept Tree

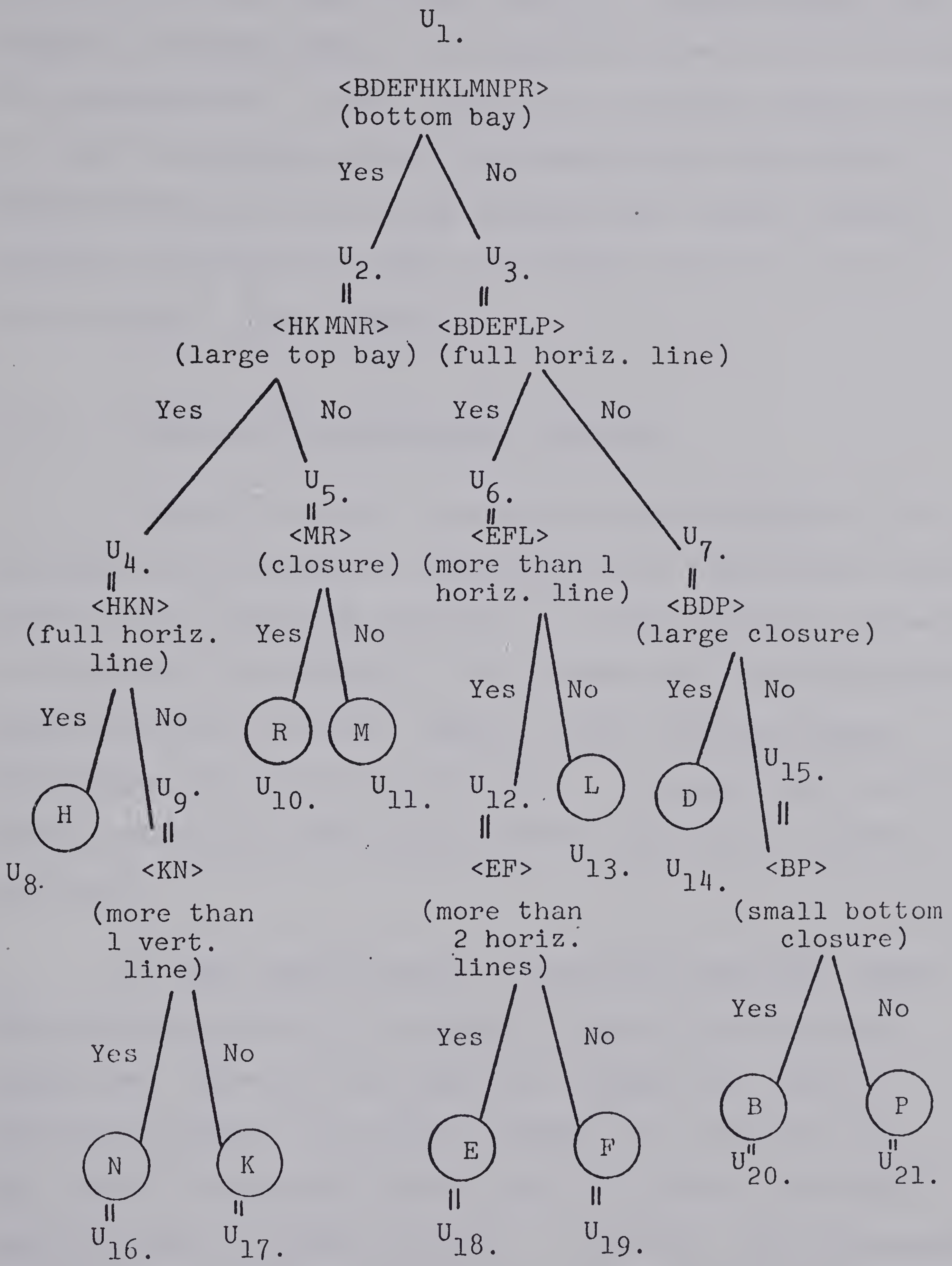


Fig. <2.2.1.2>

rule for using a name." [Hunt (1966)]. As with almost all adaptive learning models, the pre-established rule is set by the experimenter. A good selection of pattern samples should be based on quantity as well as quality for "the CLS can never be certain that it has obtained the correct concept unless it has seen at least one example of every possible description." [Hunt (1966)]

2.2.2 Generalized Discriminant Functions

Unlike the other two classification methods to be surveyed in this chapter, the generalized discriminant function method is not adaptive procedure. In this instance the entire training set is presented to the categorizer in advance of a classification decision. Based on this training sample, parameters are calculated and used to formulate the discriminant analysis for the entire pattern population (pattern universe).

In many simple pattern recognition problems linear discriminant analysis is employed in the classification algorithm. However with character recognition, where 46 different classes (26 letter alphabet, ten numerals and ten special characters) might occur, a linearly separable pattern space is highly unlikely. Therefore, this discussion will not be limited to linear discriminant functions, but will

deal with the set of generalized discriminant functions of which the linear case is an element.

In his book Decision-making Processes in Pattern Recognition, Sebestyen opens his essay on discriminant analysis by defining a metric of similarity.

> Similarity of an event v to a category is measured by the closeness of v to every one of those events $\{f_m\}$ known to be in the category. [Sebestyen (1962)].

Closeness, in the linear sense, is the mean-square distance between v and the class of patterns represented by $\{f_m\}$. In formal notation this is written as,

$$S(v,\{f_m\}) = \frac{1}{M}\sum_{m=1}^{M} d^2(v,f_m) \qquad \ldots \quad (2.2.2.1)$$

where $d^2(v,f_1)$ is the Euclidean N-space distance between the pattern point v and f_1. M is the total number of patterns representing category F.

Suppose the set of coordinates for N-space is called θ_n. It is essential to realize that the features of the events (patterns) represented by the different coordinate directions, θ_n, are not equally important in influencing the definition of the category to which like events belong. Therefore, in comparing two points (a and b) feature-by-feature, a reasonable assumption is that features with decreasing significance should be weighted with smaller weights. The idea of feature weighting is expressed by a

metric somewhat more general than the conventional Euclidean metric. That is,

$$d^2(a,b)=\sum_{m=1}^{M} (W_m(a_m-b_m))^2 \qquad \ldots \quad (2.2.2.2)$$

The feature weighting scheme, described above, is a linear transformation of the signal or pattern space.

A procedure for finding the Euclidean distance separation after a continuous "nonlinear" transformation of pattern space will now be considered. Sebestyen states that this type of operation is like "stretching and compressing a rubber sheet to bring members of a set closer to each other." [Sebestyen (1962)] In order to cluster sample vectors of a particular class, it is desired to find some invariant property of the class. Let the invariant property be some function $u(v_1,v_2,\ldots.,v_n)$ of an input vector v. Invariance is understood to mean that the function u(v) will have substantially the same value for all members of a class F, as $u_G(v)$ will have for a class G, etc. The function u(v) forms a surface over a multi-dimensional space, and it is so constructed that along the u-axis known samples of the different categories fall in disjoint intervals. See Fig. <2.2.2.1>.

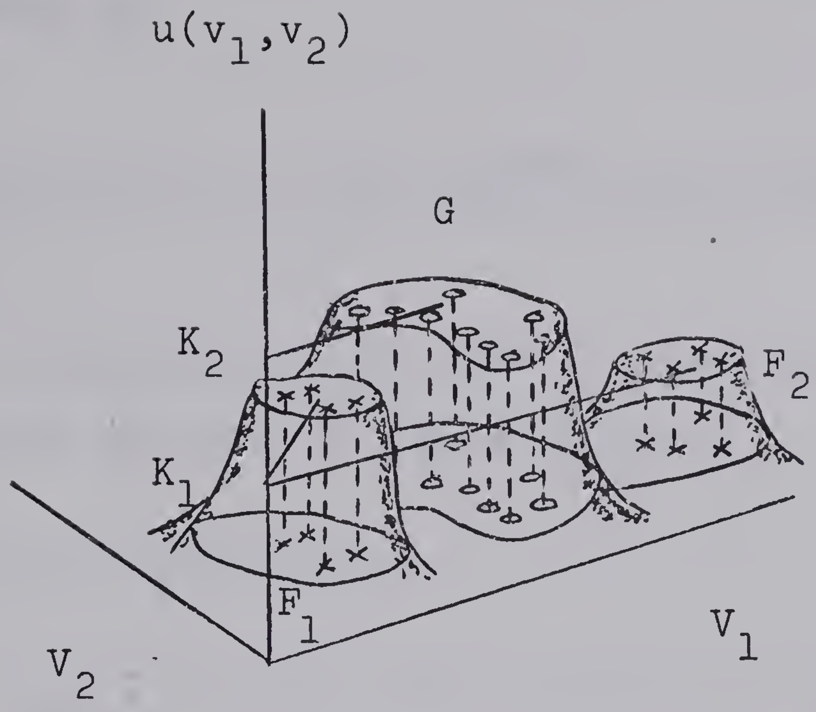


Fig. <2.2.2.1>

Transforming Pattern Space to u-space

To transform pattern space to u-space we must construct the surface u(v) so that there is a minimum variance ($\sigma_u{}^2$) about the mean over all members of the same class. The mean-square distance after this transformation is given by

$$d^2(f_m,f_n)= \frac{1}{M^2}\sum_{m=1}^{M}\sum_{n=1}^{M}[u(f_m)-u(f_n)]^2 \qquad \ldots(2.2.2.3)$$

$$=2\overline{u^2(F)} - 2\overline{u(F)}^2 = 2\sigma_u{}^2(F) \qquad \ldots(2.2.2.4)$$

In the discriminant analysis, the decision rule consists of evaluating the height of the surface over the vector, v, to be classified, and comparing this height,

u(v), with the average heights over the two sets of given samples. That is,

$$\text{decide } v\varepsilon F, \text{ if } |u(v)-\overline{u_F(v)}|<|(u(v)-\overline{u_G(v)}| \quad .(2.2.2.5)$$

where G is a category other than F. Sebestyen shows that the above criteria results in Bayes decision,

$$v\varepsilon F, \text{ if } P_F(v)>P_G(v) \qquad \ldots (2.2.2.6)$$

where $P_F(v)$ and $P_G(v)$ are the probability densities of v under the assumption that v is a member of class F or G. Also, it is assumed that the a priori probabilities of F and any G, and the cost of false alarm and false dismissal are assumed equal.

For a suitably chosen K, the general nonlinear function, u(v), can be approximated arbitrarily closely in a finite region of vector space by the following expression.

$$u(v_1,v_2,\ldots.,v_N)=\sum_{n=0}^{K}\ldots.\sum_{j=0}^{K}\sum_{i=0}^{K} a_{ij\ldots.n}{v_1}^{i}{v_2}^{j}\ldots{v_n}^{n} \quad ..(2.2.2.7)$$

Or in vector notation,

$$u(v)=\sum_{n=0}^{K} a_n\phi_n(v) \qquad \ldots (2.2.2.8)$$

Given $\phi_n(v)$ and using a method of variation, Sebestyen

is able to obtain an approximation for the coefficients of a_n.

$$\int\phi_s(v)[p_F(v)-p_G(v)]dv=\sum_{n=0}^{K} a_n\int\phi_n(v)\phi_s(v)[p_F(v)+p_G(v)]dv \quad ..(2.2.2.9)$$

Here G is chosen as the closest category to F in a multiclass environment. By letting the integral of the left-side of Eqtn. (2.2.2.9) be β_s and the integral on the right-side be $\Gamma_{ns}=\Gamma_{sn}$, the general element of the matrix Γ, Eq. (2.2.2.9) can be rewritten.

$$\beta_s=\sum_{n=0}^{K} a_n\Gamma_{ns} \qquad \text{for } s=0,1,\ldots,K \qquad \ldots \quad (2.2.2.10)$$

Using matrix notation where b is the row vector of β_s and a is the row vector of a_n we have

$$b=a\Gamma \qquad \ldots \quad (2.2.2.11)$$

The coefficients, a_n, may now be solved to yield the solution, that for a given $\phi_n(v)$, approximates u(v) with a minimum expected error. That is,

$$a=b\Gamma^{-1} \qquad \ldots \quad (2.2.2.12)$$

As an example, we may wish to approximate u(v) with a polynomial of degree K and $\phi_n(v)=v^n$. Then β_s and Γ_{ns} are moments of the probability densities as shown below

$$\beta_s = \int v^s [p_F(v) - p_G(v)] dv \qquad \ldots (2.2.2.13)$$

$$\Gamma_{ns} = \int v^{n+s} [p_F(v) + p_G(v)] dv \qquad \ldots (2.2.2.14)$$

Remarks

While many steps have been bi-passed in the treatment of this classification method, it is easy to appreciate the complexity involved in calculating a generalized discriminant. However, if prior knowledge of the probability densities of each category is available, this method may be applied to achieve an optimum* machine. Then, undoubtedly this classification rule would supercede those rules developed elsewhere in this chapter. Unfortunately this type of information is rarely available.

2.2.3 Piecewise Linear Discriminant Functions.

In Piecewise Linear (PWL) classifications a non-parametric training method is employed to create prototype points which are utilized in the discriminant analysis. The

*If the course of action chosen by the categorizer has the smallest expectation of error then the rule that is forwarded is termed "optimum". An optimum machine is achieved when the unknown sample values are random variables with known probability distributions. An optimum machine is also called a Bayes machine.

use of prototype points not only facilitates discriminant computation, but also alleviates the necessity of storing the entire training set. PWL functions are calculated by applying linear discriminant analysis to a special adaptive classification algorithm-an algorithm that is discussed later in this section. First, however, linear discriminant functions are introduced, since they are the basis of the PWL case.

In his book Learning Machines , Nils Nilsson develops a classification technique from a metric based on minimum Euclidean distance (M.E.D.) measurements.

> Associated with each point P_i is a category number i=1,2,...,k. A minimum-distance classifier, with respect to the points $P_1, P_2, \ldots, P_k$, places each pattern vector X into that category i_c which is associated with the nearest P_i of the points $P_1, P_2, \ldots P_k$. [Nilsson (1965)]

The points $P_1, P_2, \ldots, P_k$ are called prototype points.

An equivalent classification can be obtained by comparing the squared Euclidean distances and finding the minimum for i=1,2,...,R categories.

$$\min_{i=1,\ldots,R} d^2(X,P_i) = |X-P_i|^2 = X.X - 2X.P_i + P_i.P_i \quad \ldots(2.2.3.1)$$

By multiplying Eq. (2.2.3.1) by -1/2, dropping the

X.X term (since it is present for all decisions) and finding the maximum value for $g_i(x)$, the metric stated in Eq(2.2.3.1) is equivalent to Eq. (2.2.3.2). That is,

$$g_i(X)=X.P_i-1/2P_i.P_i \qquad \ldots\ (2.2.3.2)$$

From this expression the minimum-distance classifier is observed to be a linear machine in X. The decision surface between two adjacent regions, R_i and R_j, is a hyperplane described in Eq. (2.2.3.3)

$$g_i(x)-g_j(x)=T \qquad \text{for all } j\neq i \ \ldots(2.2.3.3)$$

where T is usually set to zero (Bayes Rule)

Fig. <2.2.3.1> shows the decision regions for a minimum distance classifier with respect to the points P_1,P_2 and P_3.

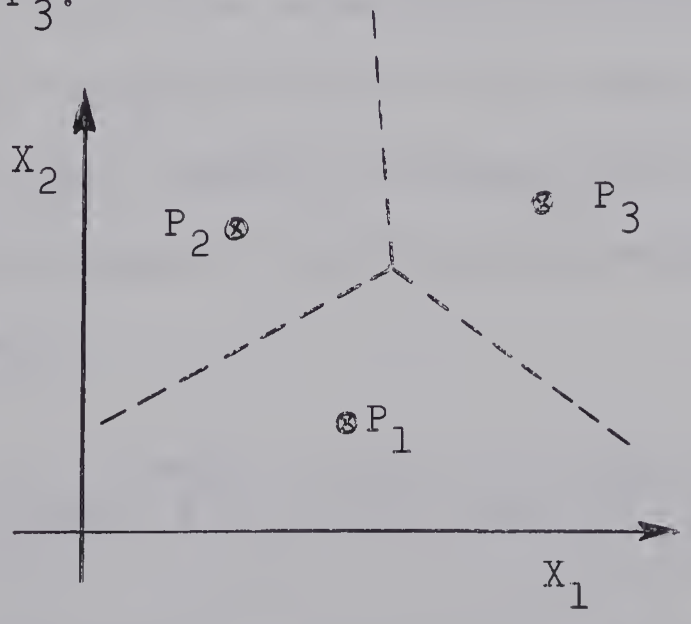


Fig. <2.2.3.1>

Like linear machines, PWL machines also perform minimum distance classifications with respect to a finite number of point sets (or prototype vectors). However, PWL discriminant functions differ from the linear discriminant functions because more than one prototype vector may belong to the point set which represents that category. To be more specific, assume there exists R finite point sets $P_1, P_2, \ldots,$ P_R. Then,for each i=1,...,R, let the ith point set consist of L_i prototype points $P_i^{(1)}, P_i^{(2)}, \ldots, P_i^{(Li)}$

A metric for the PWL case may be defined as

$$d^2(X,P_i) = \min_j |X-P_i^{(j)}|^2 \qquad \ldots (2.2.3.4)$$

where $d(X,P_i)$ is a measure of Euclidean distance in d-space. Proceeding as in the linear discriminant case, the r.h.s. of Eq.(2.2.3.4), is expanded and multiplied by -1/2. The X^2 term is dropped since it is common to all discriminants. The metric for finding a minimum now becomes a problem of finding a maximum. The PWL discriminant function is written as follows:

$$g_i(X) = \max_j \quad P_i^{(j)} \cdot X - 1/2 P_i^{(j)} \cdot P_i^{(j)} \qquad \ldots (2.2.3.5)$$

Like the linear case, the decision surface for PWL functions consists of sections of hyperplanes described by

Eq.(2.2.3.5). In an effort to place a rejection bound* on the discriminant, Syms [1967] suggests the formation of hyperspheric discriminant surfaces. These surfaces are formed by re-introducing the X^2 component that is dropped from the original PWL metric (Eq.(2.2.3.4)). By adding to the X^2 term to a threshold value, T, a new decision is created.

$$\max_i g_i(X)>T+1/2X^2 \text{ , } i=1,...,R \qquad ... (2.2.3.6)$$

That is, X is placed in category i if $g_i(X)$ is the maximum PWL discriminant over all R categories, and if it satisfies the criteria that it be larger than $T-1/2X^2$. By expanding Eq.(2.2.3.6), it is observed that T is a measure of the radius of a hypersphere.

$$g_i(X)= \max_k X.P_i^{(k)}-1/2(P_i^{(k)})^2>T+1/2X^2 \quad ... (2.2.3.7)$$

$$\text{which implies } |T|\leq 1/2d^2(X,P_i^{(k)}) \qquad ... (2.2.3.8)$$

The threshold value, T, is a measure of the cost of making

---

*The rejection bound as formulated by Syms is based on a set of curve parameters which are characteristic of the particular application (emission spectra of a nuclear reactor). However with a character recognition model, no such parameter exists. Therefore a nearest neighbor procedure was developed which gives a rejection bound that is independent of the application.

one of two kinds of decision errors: false alarm and false dismissal. The probability of false alarm is increased if T is made too large. The probability of false dismissal is increased if T is made too small. Under the assumption that the two types of errors are equal, the decision hyperplanes in Eq.(2.2.3.5) should be formed. That is, $T=g_j(x)-1/2X^2$ where $j\neq i$. However, in some problems (in particular the character recognition problem) the cost of making an error of false dismissal is less than the cost of making an error of false alarm. Chapter IV contains further information extending the threshold theory mentioned in this section.

There are two major problems with training PWL machines.

1. A method must be found which appropriately adjusts the subsidiary discriminant weights.

2. A method must be found which appropriately creates and/or destroys subsidiary discriminants (subcategories).

There is two methods by which the pattern classifier adjusts subsidiary discriminant weights. One method, called <u>error-correction</u> training, makes corrections to the category weight vector only when a pattern is misclassified. The other method, called <u>modeseeking</u> training, adjusts the

weight vectors for every pattern and attempts to locate the prototype vector at a pattern cluster center. The error-correction training [Nilsson (1965)] moves the separating hyperplane (or hypersphere) between the pattern and the correct category enough to insure that the present input vector is placed in the correct category. The formula for this procedure is given by

$$P_i' = \beta P_i + (1-\beta)X \text{ where } \beta \leq \frac{|X|^2 - P_k . X}{|X|^2 - P_i . X} \quad \ldots(2.2.3.9)$$

For the mode-seeking training method (as proposed by Stark, Okajima and Whipple (1962)) Eq.(2.2.3.9) is used, but the betafactor is set in Eq.(2.2.3.10).

$$\beta = n/_{(n+1)} \qquad \ldots (2.2.3.10)$$

where n equals the number of times this prototype vector, Pi, has been adjusted previously.

The training procedure for a PWL machine is as follows. A set of training patterns, X_k,k=1,...,m are sequentially presented to the machine for t iterations. The parameter t is set heuristically, and likely depends on the number of training patterns and number of mistakes made by the machine in the previous iteration. During these initial t-iterations, the machine attempts to fit the

patterns into categories as a linear machine would. At the completion of t iterations, if patterns are still incorrectly classified (implying nonlinear separability), then subcategories are formed. A prototype weight vector is stored for each new subcategory created. In discriminant calculations a category which is composed of many subcategories is represented by its closest subcategory to the input pattern vector. The classification rule is as described in Eq. (2.2.3.3) where T=0.

The drawback of this training procedure is the creation of a multitude of small subcategories which implies large storage requirements. However, provisions may be made to delete subcategories if they fall into disuse. Rosen and Hall [1966] developed a model which has this capability.

Remarks

Of the three nonlinear classification methods surveyed in the chapter, the PWL discriminant method is found most often in the literature. Munson [1968], Rosen and Hall [1966], Duda and Fossum [1966], and Syms [1967] all achieved excellent results with PWL discriminant functions. Some of these authors employed mode-seeking or error-correction training only, while others combined the two methods. However, improvements are needed in the training algorithm sc that as near an optimum solution as

possible is achieved. In Chapter IV a new training algorithm is explored which converges to a solution within a given number of steps. Some authors (Syms [1967], Ball and Hall [1964]) suggest that a unique threshold be available for each subcategory. An extension and implementation of this idea is discussed in Chapters III and IV and forms an essential part of the research in this thesis.

2.2.4 Concluding Remarks

This chapter has surveyed three methods of solving a non-linear pattern recognition problem. Very little is known about the performance of the CLS and the Generalized Discriminant method; simply because results from a good implementation are not available. Therefore, what is stated about these methods is conjecture on the author's part.

Given the proper set of conditions both the CLS and the Generalized Discriminant method yield more accurate classification rules than the PWL system. If superior preprocessing is received in which few, if any, extraneous features are extracted from the input pattern, then the CLS should provide an optimal classification. The main pitfall of the CLS is its tendency to center its classification rule about irrelevant features leaving large groups of patterns indistinguishable. The result of such a strategy is misclassification and the need to continually recalculate

the entire decision structure.

If the probability distribution (multi-variate normal, poisson, Bernoulli, etc.) is known prior to classification, then Sebestyen's Generalized Discriminant technique provides an optimum or near optimum classification rule. However, in most practical problems this a priori information is not available or varies for each category created.

The PWL classification method is fast, accurate (although by no means optimum) and easy to implement. As well, this classification algorithm is chosen for our categorizer because of the ease in which it can be adapted to the Fourier transform-type preprocessor. Theory resulting from this research on trainable classifiers is presented in Chapter III. The simulation and a possible implementation of the categorizer which performs surface-fitting classifications is outlined in Chapter IV.

CHAPTER III

THE THEORY OF HPRS

In Chapter III, some theoretical developments of the research outlined in Chapter II are presented. The theory of the optical (Fourier) transform is discussed in Sec. 3.1. Sec. 3.2 describes the extensions that are made to the theory of PWL classifiers.

3.1 The Theory of the Optical Transform

The concepts of coherent optical transformation of spatial frequencies can be explained with reference to Fig. <3.1.1>.

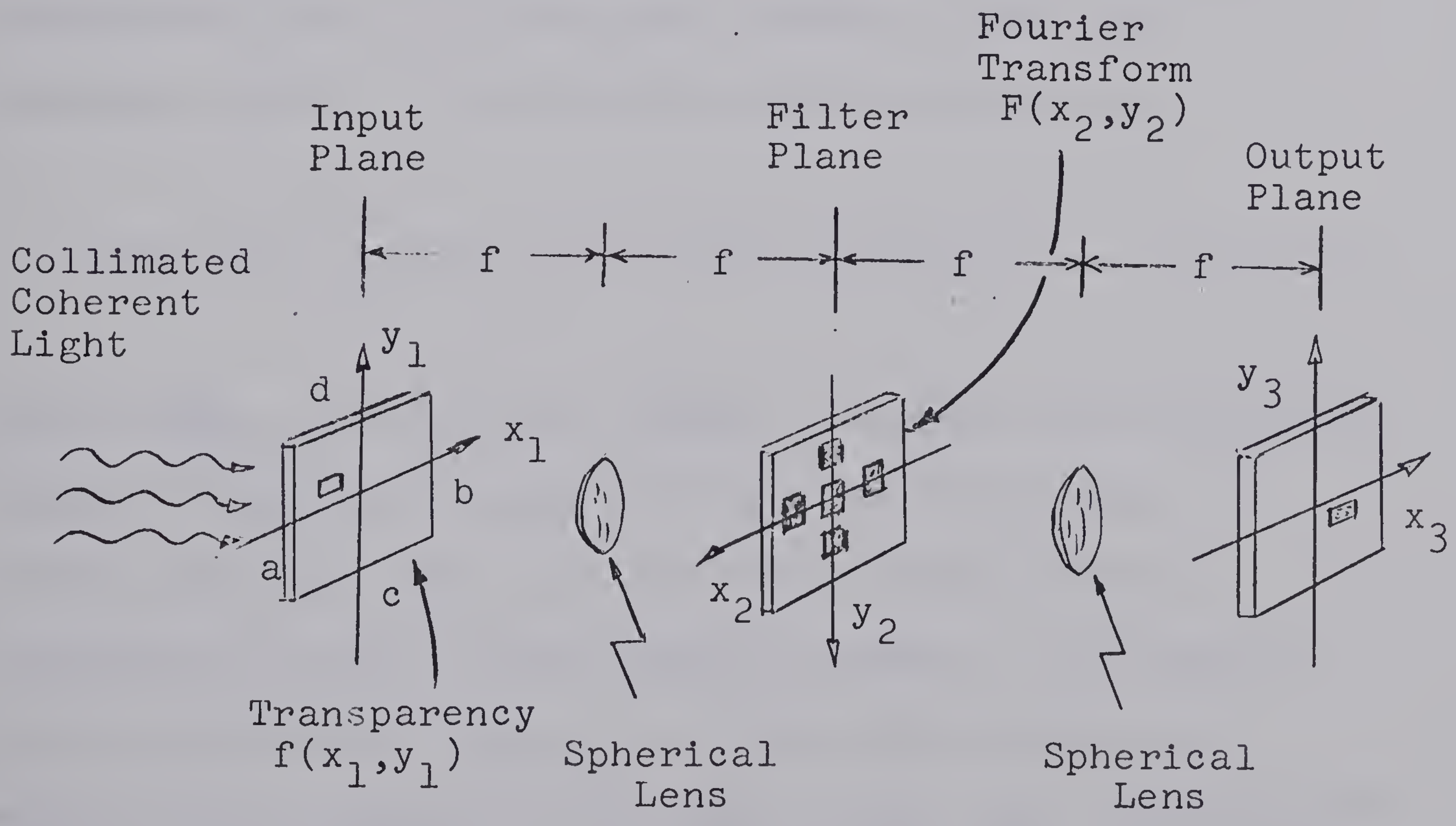


Fig. <3.1.1.> Optical Transform

A transparency in the x_1, y_1 plane with transmittance $f(x_1, y_1)$ is illuminated by a coherent, collimated light beam from a laser. At the input plane the electric field amplitude of the light is proportional to $f(x_1, y_1)$. The first spherical lens produces an image of the transparency in the filter (transform) plane. The light amplitude, $F(x_2, y_2)$, in the filter plane, is determined by the Kirchhoff integral of diffraction theory, and given by the Fourier transform relation.

$$F(x_2,y_2) = \int_c^d \int_a^b f(x_1,y_1) e^{\{\frac{2\pi i}{\lambda f}(x_1x_2+y_1y_2)\}} dx_1 dy_1 \quad \ldots \quad (3.1.1)$$

λ is the wavelength of the light illuminating the transparency and f is the focal length of the lens [Andrews (1968)]. The relation can be written as

$$F(u,v)=\int_c^d \int_a^b f(x_1,y_1) e^{\{i(ux_1+vy_1)\}} dx_1 dy_1 \quad \ldots \quad (3.1.2.A)$$

where $u=\frac{2\pi}{\lambda f}x_2$ (3.1.2.B) and $v=\frac{2\pi}{\lambda f}y_2$ (3.1.2.C) are called the spatial frequencies in the Fourier transform plane. A second spherical lens will perform a second Fourier transform to return to the spatial domain. The electric field distribution, $g(x_3, y_3)$ in the output plane is a replica of the object at the input plane, but rotated 180°

about the center axis.

$$g(x_3,y_3)=F\{F(u,v)\} \qquad \ldots (3.1.3)$$

The fact that the original image may be regenerated in above manner has some important applications. First, the accuracy of the simulation program for an optical transform can be checked by using the two-dimensional Fast Fourier Transform (FFT) procedure. Secondly, the process of optical holography uses this phenomenon to create a three-dimensional effect from a two-dimensional object.

F(u,v), as described in Eq.(3.1.2), is found by evaluating the double integral of a continuous transmittance function $f(x_1,y_1)$. This form is not conducive to computer simulation for the transmittance function, $f(x_1,y_1)$, is almost never known and therefore is approximated by a matrix representing discrete points on the plane. Assuming f(x,y) is a two dimensional array of points of dimension N by N, the two dimensional Fourier transform F[u,v], in discrete form, is defined as

$$F[u,v]=\frac{1}{N}\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} f(x,y)\exp\{\frac{2\pi i}{N}(xv+yu)\} \qquad \ldots (3.1.4)$$

The evaluation of equation (3.1.4) can be broken into two steps. First, the one dimensional Fourier

transform, F[u,y], is taken along the x coordinate of every horizontal line of f(x,y). Then a second dimensional Fourier transform is taken in the y direction of every vertical line of F[u,y] to yield a composite transform, F[u,v].

At this point an analysis of the content of a matrix F, which contains the discrete value of F[u,v] will be made. With reference to Fig. <3.1.2> an appropriate question to ask would be "What does the $F_{1,4}$ element represent?"

Original Letter I

0	0	1	0
0	0	1	0
0	0	1	0
0	0	1	0

FFT of the Letter I

Spatial Freq. Harmonics

Matrix Coordinates

Transform Values

	0	1	-2	-1
0	1,1 1	1,2 1	1,3 1	1,4 1
1	2,1 0	2,2 0	2,3 0	2,4 0
-2	3,1 0	3,2 0	3,3 0	3,4 0
-1	4,1 0	4,2 0	4,3 0	4,4 0

Fig. <3.1.2>. FFT of the Letter I

The $F_{1,4}$ element represents the spatial frequency (as given by equations 3.1.2.B and 3.1.2.C) which makes one cycle across the width and four cycles across the height of the transform image. As an example, suppose the character I (single vertical bar) is presented to the preprocessor. If the preprocessor output is a 4x4 matrix, the result would be similar to the matrix in Fig. <3.1.2>. Hence the only

nonzero values to appear are for the elements $<F_{11}, F_{12}, F_{13}, F_{14}>$ -the zero order frequencies in the horizontal direction.

From the theory that is developed in this chapter the Fourier transform can be viewed as a measure of the frequency domain of a character. A spatial representation of the object is transformed into a set of discrete values which denote the relative spacing of hills (where a line is present) and valleys (where a line is not present) on the character "terrain".

3.2 Extending the Theory of PWL Classifiers

In this section the theory of PWL classifiers is extended in three areas. First a new discriminant function, $g_i(x)$, which is based on a nearest neighbor, "full" Euclidean metric is formulated. Second, some new concepts in threshold theory are developed which evolve from the new discriminant technique. Included in this development is a discussion of three different subcategory operations. Third, an adaptive training algorithm is presented that converges to a solution within a given number of iterations. The algorithm attempts to minimize the number of subcategories that are created while producing a categorizer that can identify the training set correctly. In Sec. 3.3.5 two methods for classifying

unknown patterns from the environment (those not belonging to the training set) are discussed.

3.3.1 Discriminant Function

To accommodate Syms' [1967] suggestion of a rejection threshold for each category, the X^2 term is reintroduced in Eq.(2.2.3.2). The form of the discriminant is now written as,

$$g_i(X) = \max_j \left(P_i^{(j)}.X - 1/2P_i^{(j)}.P_i^{(j)} - 1/2X.X \right) \ldots \quad (3.2.1.0)$$

That is, $g_i(X) = -1/2d^2(X, P_i^{(j)})$, where the d^2 function is the square of the measure of the Euclidean distance in N-space. More appropriately these discriminants shall be called Piecewise Quadric (PWQ) discriminants functions since $g_i(X)$ is quadratic in X. The quadric surface formed by $g_i(X)$ is a hypersphere with radius equal to $2\,|g_i(X)|^{1/2}$. The larger $g_i(X)$ becomes, the further X is from $P_i^{(j)}$ in Euclidean N-space measure. If a rejection threshold is set at $T_i^{(j)}$, then a hyperspheric rejection surface is formed by the following rejection rule,

decide $X\varepsilon$ category i if,

$$g_i(X) > T_i^{(j)} \qquad \ldots \quad (3.2.1.1)$$

otherwise reject X. In the next section the method for calculating $T_i^{(j)}$ shall be described.

The decision surface formed between neighboring categories, however, remains a hyperplane since our classification rule is linear in X. That is,

decide $X \varepsilon$ category i if,

$$g_i(X)-g_k(X)>0 \qquad \ldots (3.2.1.2)$$

for all categories $k \neq i$.

The linear relation in X can be shown by expanding the l.h.s. of Eq.(3.2.1.2) and substituting the r.h.s. of Eq.(3.2.1.9) for $g_i(X)$ and $g_j(X)$.

$$\max_i \left(P_i^{(j)} \cdot X - 1/2 P_i^{(j)} \cdot P_i^{(j)} \right) - \max_k \left(P_k^{(m)} \cdot X - 1/2 P_k^{(m)} \cdot P_k^{(m)} \right) > 0 \qquad \ldots (3.2.1.3)$$

Therefore our decision rule is made up of a quadratic rejection rule and a linear classification rule. That is,

decide $X \varepsilon$ category i if

$$g_i(X) > T_i^{(j)} \ \underline{\text{and}} \ g_i(X) - g_k(X) > 0 \qquad \ldots (3.2.1.4)$$

for all categories $k \neq i$.

Fig. <3.2.1> shows a decision surface that is constructed

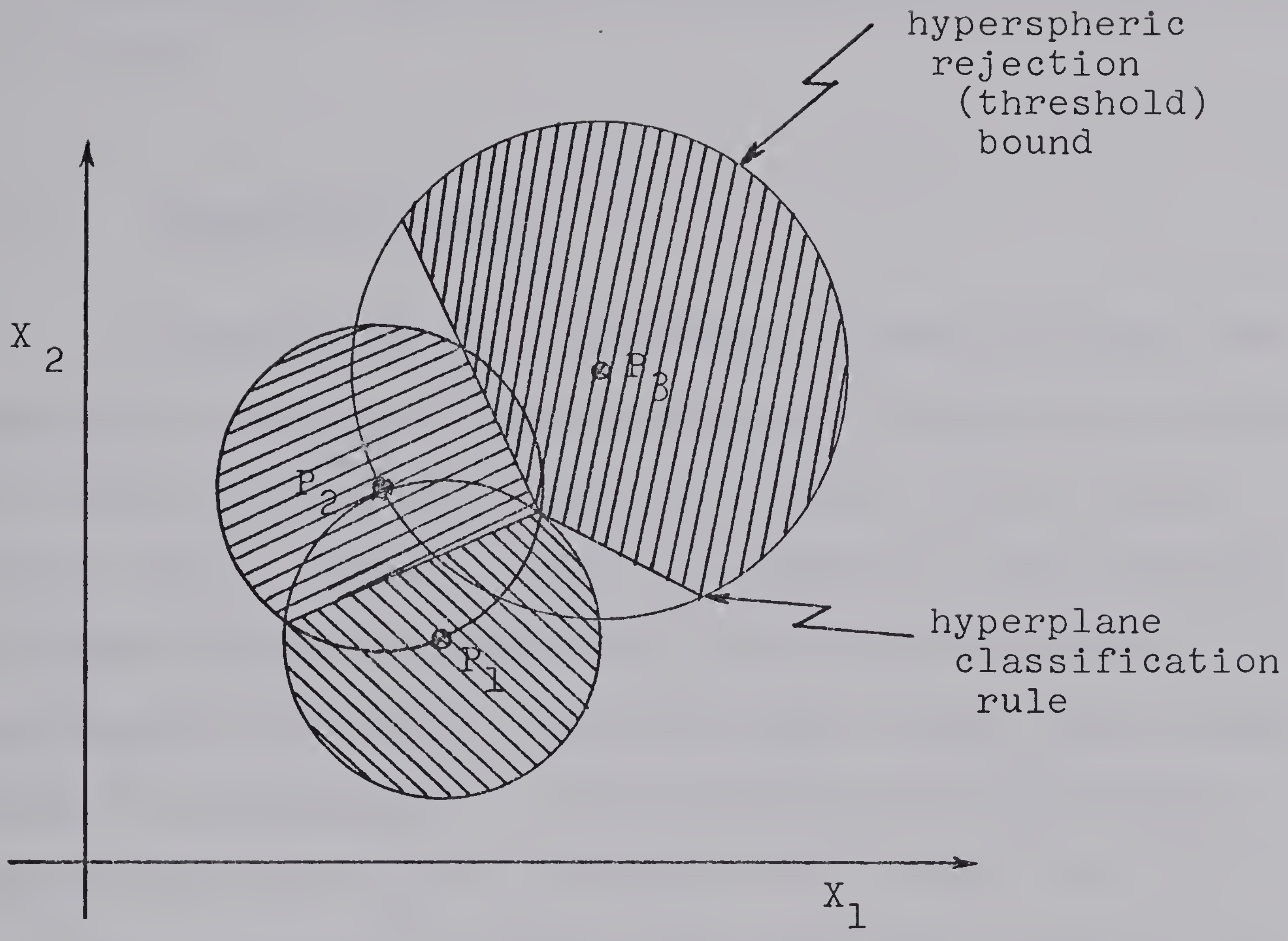

hyperspheric rejection (threshold) bound
X_2
P_3
P_2
P_1
hyperplane classification rule
X_1

Fig. <3.2.1>

Decision Surface for Three Category Recognition Problem in Two Dimensional Pattern Space.

for a three category recognition problem in two dimensional pattern space. The topology of these surfaces can be described as sliced circles. For the more general case - m categories in n dimensional space-sliced, hyperspheric decision surfaces are created.

3.2.2 Threshold Theory

By employing PWL discriminant classifications (as described by Nilsson [1965]) categories* and/or subcategories are formed by hyperplane decision surfaces which disect pattern space. Many of these subcategories are unbounded or geometrically structured such that points outside a subcategory are closer to the prototype vector than points within the subcategory. By redefining the discriminant function and letting the decision rule be Eq.(3.2.1.4), it is now necessary to center the prototype vector in the middle of the subcategory. Therefore no pattern is allowed to be a member of a subcategory unless its pattern point is within a threshold distance, $T_j^{(i)}$, of the centered prototype

---

*Even though a category may be composed of a single subcategory, in the discussion to follow only the term subcategory shall be used where both terms may be applicable. To clarify notation $T_j^{(i)}$ means the threshold value for the i subcategory of category j.

vector. Let $T_j^{(i)}$ be set proportional to the Euclidean N-space distance between the prototype vector of the subcategory $C_j^{(i)}$, and the prototype vector of the nearest subcategory, $C_k^{(m)}$ (nearest neighbor technique). That is,

$$T_j^{(i)}(P_k^{(m)})=\delta\left(P_j^{(i)}\cdot P_k^{(m)}-1/2P_j^{(i)}P_j^{(i)}-1/2P_k^{(m)}P_k^{(m)}\right) \qquad \ldots \quad (3.2.2.0)$$

where δ is the nearest neighbor weighting factor.

$T_j^{(i)}$ is a measure of the cost of making an error of false dismissal or an error of false alarm when classifying X. The smaller $T_j^{(i)}$ is made, the greater is the probability of committing an error of false dismissal. The larger $T_j^{(i)}$ is made, the greater is the probability of committing an error of false alarm. By setting the δ-factor to 1.000 (which was what occurred during the thesis research), the threshold $T_j^{(i)}$ is set equal to the distance, in discriminant measure (i.e. $-1/2d^2(P_j^{(i)},P_k^{(m)})$), between $P_j^{(i)}$ and $P_k^{(m)}$ where $C_k^{(m)}$ is the closest subcategory to $C_j^{(i)}$. This appears to be an appropriate estimate of the cost of making either one of the two types of decision errors.

3.2.3 Operations on Subcategories

A. Birth of a Subcategory

A subcategory is created if the categorizer decides

that a pattern can not be placed into any of the existing subcategories. The decision as to whether a pattern does or does not fit into any of the present subcategories changes throughout the training period. When the training algorithm is discussed later in this chapter, these changes are mentioned.

B. Submerging a Subcategory

The nearest neighbor to a subcategory, $C_j^{(i)}$, may be subcategory $C_j^{(k)}$, which also belongs to the category j. In such a situation it would seem unreasonable to use a nearest neighbor calculation for $T_j^{(i)}$. Instead, a search is made to find the closest subcategory that does not belong to category j. Once found, this subcategory is used in the nearest neighbor threshold calculation. A subcategory is said to be submerged in $C_j^{(i)}$, if it belongs to category j and is closer than the nearest subcategory which does not belong to category j.

A maverick from a foreign category may also be submerged if it is the nearest neighbor to a subcategory which possesses two special qualities:

1) The subcategory (call it $C_j^{(i)}$) must contain a member that is a greater distance away from $P_j^{(i)}$ than the maverick.

2) This particular member must satisfy the "death" conditions for subcategory $C_j^{(i)}$.

C. Death of a Subcategory

Subcategories are only eliminated after they become ineffectual in the discriminant analysis. A subcategory $C_j^{(i)}$ is ineffectual if it satisfies the following three death conditions*:

1) The subcategory $C_j^{(i)}$ contains only one member of the learning set.

2) The subcategory is within a discriminant distance $T_j^{(k)}$ of subcategory $C_j^{(k)}$.

3) Both subcategory $C_j^{(k)}$ and $C_j^{(i)}$ have the same nearest neighbor subcategory.

Fig. <3.2.3.1>, <3.2.3.2> and <3.2.3.3> pictorially demonstrate situations in which subcategories are created, submerged, and destroyed.

3.2.4 Training Algorithm

The training algorithm shall be described by relating the adaptive strategy that is involved during each

---

*It was decided strictly on a heuristic basis to limit the number of subcategories that are formed. If $C_j^{(i)}$ satisfies the three conditions listed, it possesses almost the same influence on a classification decision as $C_j^{(k)}$-therefore $C_j^{(i)}$ is eliminated.

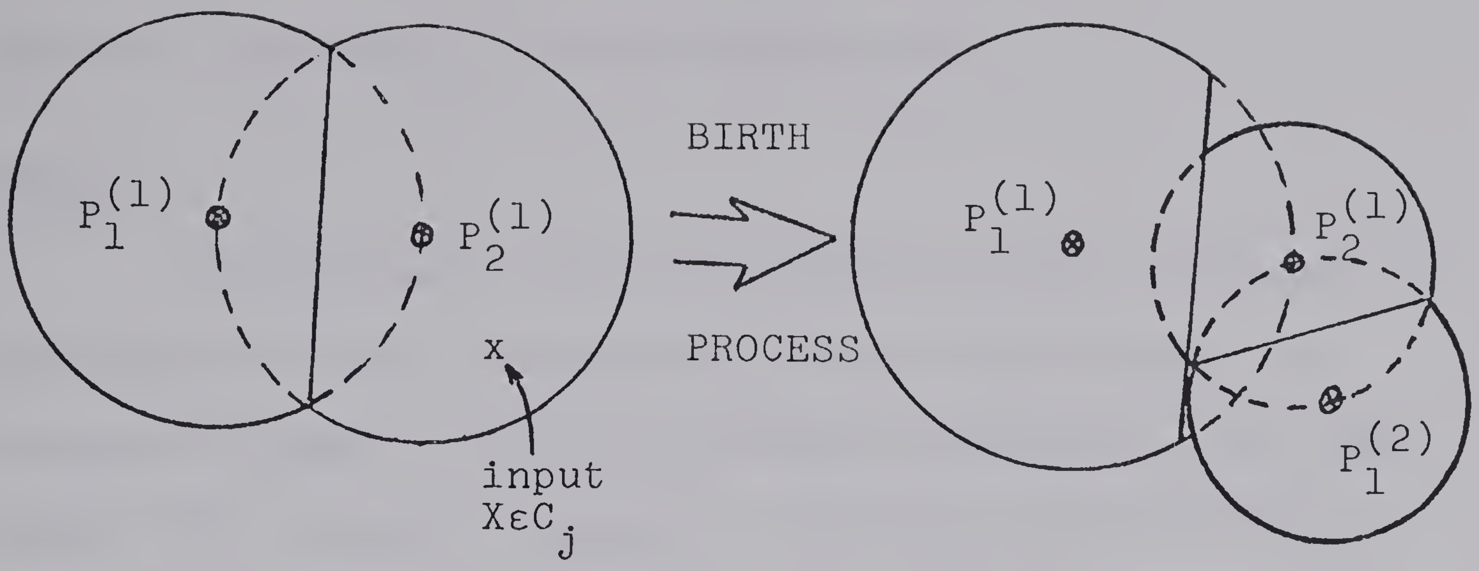


Fig. <3.2.3.1>

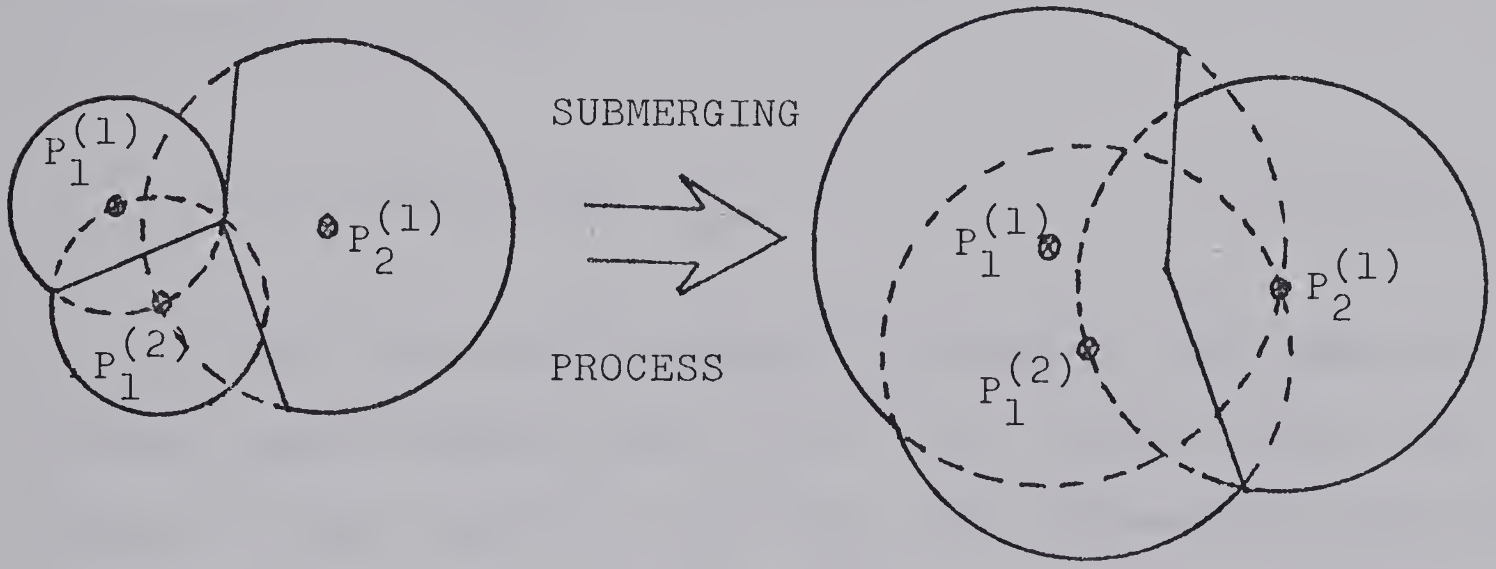


Fig. <3.2.3.2>

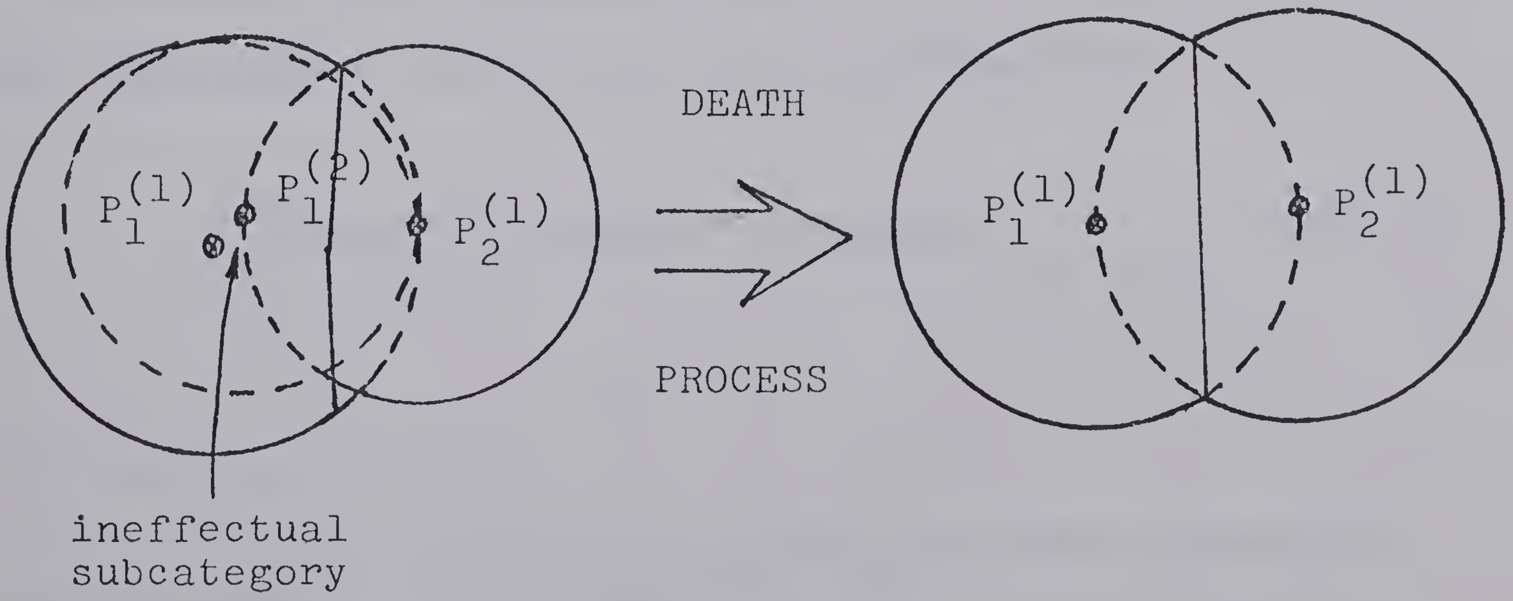


Fig. <3.2.3.3>

pass (or iteration) of the training set.

Pass I

During Pass I of the training set the patterns are introduced to the categorizer for the first time. An attempt is made to fit all patterns belonging to the same class into a single category. To help cluster patterns belonging to the same category, an initial threshold, T', is set by Eq.(3.2.4.0)

$$T_j'^{1}(X,\gamma)=\gamma . X^2 \qquad \ldots (3.2.4.0)$$

X is an input pattern belonging to category j and gamma is a factor which ranges from 0 to 1.[*] If X falls within the threshold bound (that is, $g_j^{(X)} \geq T_j$) then modeseeking training is applied (Eq.(2.2.3.10)). If X falls outside the threshold range ($g_j^{(X)} < T_j$) the prototype vector, $P_j^{(i)}$, is adjusted so X lies within the threshold bound. The formula for making the appropriate error-correcting adjustment is

$$P_j'^{(1)}=\beta P_j^{(1)'}+(1-\beta)X \qquad \text{where} \qquad \ldots (3.2.4.1)$$

[*] In the simulation program excellent clustering resulted when γ is set equal to .200.

$$\beta = \left[\frac{T_j'^{(1)}}{g_j(X)}\right]^{1/2} \qquad \ldots (3.2.4.2)$$

If X is the first pattern that is introduced belonging to category j, then $T_j'^{(1)}$ is set to zero, implying $P_j^{(1)}=X$. At the end of Pass I and all other iterations, a three element set of nearest neighbor subcategories are found for each subcategory.*

Pass II

Pass II continues category training. That is, an attempt is still made to fit all patterns belonging to the same class into a single category. The threshold value function, used in this iteration and all further iterations, is the nearest neighbor calculation in Eq.(3.2.2.0). In all cases the δ-factor is set equal to 1.000. If the pattern X falls within $T_j^{(1)}$ $(g_j(X) \geq T_j^{(1)})$, then no adjustments are

---

*Because this method of classification is so dynamic (i.e. the prototype vectors are being continually moved about pattern space) recording only the nearest neighbor will not suffice for threshold calculation. Instead the three closest subcategories are recorded. When a threshold value is required, calculations are made using the prototype vectors of all three subcategories. The minimum value of the three calculations is set as the threshold.

made to $P_j^{(1)}$. If X fall outside $T_j^{(1)}(g_j(X)<T_j^{(1)})$, then error-correcting adjustments are made as designated by Eq.(3.2.4.1). In this instance the β-factor is calculated using Eq.(3.2.4.3).

$$\beta = \left[\frac{T_j^{(1)}}{g_j(X)}\right]^{1/2} \quad * \qquad \ldots (3.2.4.3)$$

Pass III

Thus far the categorizer has treated the classification problem as though pattern space is linearly separable. If pattern space is linearly separable, it is hoped, by Pass III, that the categorizer has stabilized-implying that it is capable of correctly classifying the training set. If pattern space is nonlinearly separable or the categorizer has not stabilized for the linear case, then subcategories must be created (BIRTH PROCESS). Or, to be more specific, if for some category j

1) $g_j(X)<T_j^{(1)}$
2) $g_j(X)-g_k(X)<0$ for any category $k\neq j$ and
3) a subcategory, $C_j^{(2)}$, for category j has not been

----

*The derivation of the one-step error correction β factor is developed in Appendix A.

created previously,
then a new subcategory, $C_j^{(2)}$, is created.

Once a new subcategory is created for category j, no further subcategories may be formed for that category in this pass. An attempt is made to fit the training patterns belonging to j into two subcategories rather than one.

At the completion of Pass III subcategories are checked for submergence or deletion as specified in the section on "Subcategory Characteristics". This procedure plus the calculation of the sets of nearest neighbor categories must be completed at the end of this and all further passes.

Pass IV And Upwards

Pass IV, Pass V, etc. are all based on the same principles as outlined in Pass III. In Pass IV the greatest number of subcategories belonging to any category is three-(Pass V-four, and on up). Hence, the categorizer's strategy is to discover a solution in the fewest number of iterations thus creating the fewest number of subcategories. An obvious limit to the number of iterations that are required before a solution to the training set is formed is the number of patterns in the training set. That is, the trivial case

occurs when each pattern is the prototype vector of a separate subcategory.

3.2.5 Pattern Classification

Patterns are classified on the basis of the categorizer's state at the completion of the training period. Therefore, the main assumption behind any classification scheme is that the categorizer has been trained on a learning set which adequately represents the pattern environment.

Two different methods of classification can be used:

1) Absolute MED Classification.

The input pattern X is classified as belonging to the category which contains a subcategory that is the minimum Euclidean distance from X of all the existing subcategories. Under MED classification no patterns are rejected - all are classified.

2) Relative Reliability Classification

In relative reliability classification, threshold measures are used rather than Euclidean distance measures. A threshold measure is defined as

$$TM_i^{(j)} = \frac{g_i(X)}{T_i^{(j)}} \qquad \ldots (3.2.5.0)$$

Where $T_i^{(j)}$ is found by Eq.(3.2.2.0) and $g_i(X)$ is found by Eq.(3.2.1.0). A pattern X is classified as belonging to category i if $TM_i^{(j)}$ is the least threshold measure over all subcategories and is less than a classification rejection bound (CRB). Typically, CRB=1 since this is the value on which the machine is trained.

3.3 Concluding Remarks

In this chapter the theory of the optical transform was presented. The aim of this theoretical development was to acquaint the reader with the fundamental concepts which relate to Fourier transform-type preprocessor, PREP. Similarily, the heuristical extensions to the theory of PWL machines are applied to the pattern categorizer LEM. With Chapter III as the basic guideline, attention will now be focused on Chapter IV which outlines a possible hardware implementation and the software simulation of the HPRS model.

CHAPTER IV

A DESCRIPTION OF HPRS

In this chapter HPRS is formally defined as a finite state device. The remainder of the chapter is devoted to a description of the hardware implementation and the program simulation of preprocessor stage (PREP) and the categorizer stage (LEM) of HPRS.

4.1 Machine Definition

Recently, in pattern recognition literature, there has been a trend towards defining the machine characteristics of the pattern recognition device. This trend has evolved for two reasons. One, a machine definition is interesting because it supplies a basis by which two or more machines may be compared. Two, a machine definition is beneficial because there exists special algorithms which utilize the definition to minimize the machine. Since the pattern recognition device described in this research is almost an infinite state machine*, a machine description shall be

---

*For example, let the prototype points stored in the categorizer be represented by twenty, nonbinary (0-9), 16 - component vectors. Then the categorizer, alone, would have $(16x10)^{20}=160^{20}$ possible states. Present minimization methods (such as using a state transition table) would be to unwieldy to make any useful state reduction.

presented out of interest only.

A field of study, Automata Theory, has been developed which uses special set theory nomenclature to formalize a machine definition. This special notation shall be employed in the machine description that follows. First, however, some basic definitions shall be presented.

Let $\Sigma=\{\sigma_1,\sigma_2,....,\sigma_n\}$ and $B=\{b_1,b_2,....,b_m\}$ be finite nonempty sets of symbols. A sequential machine over Σ (the input alphabet) and B (the output alphabet) is a quadruple

$M=<S,M,s_o,N>$

where S is a finite set of internal states,

M is the state transition function such that $S\times\Sigma \overset{M}{\rightarrow} S$,

s_o is the output transition function.

A sequential machine is called a MOORE MACHINE if the output depends only on the present state of the machine such that $S\overset{N}{\rightarrow}B$. Fig. <4.1.1> shows a block diagram of a Moore machine.

HPRS Definition

The HPRS preprocessor, PREP, acts as a transfer function. It maps an input pattern Y into a spatial frequency representation X ($Y\xrightarrow{PREP}X$). The mapping is one-

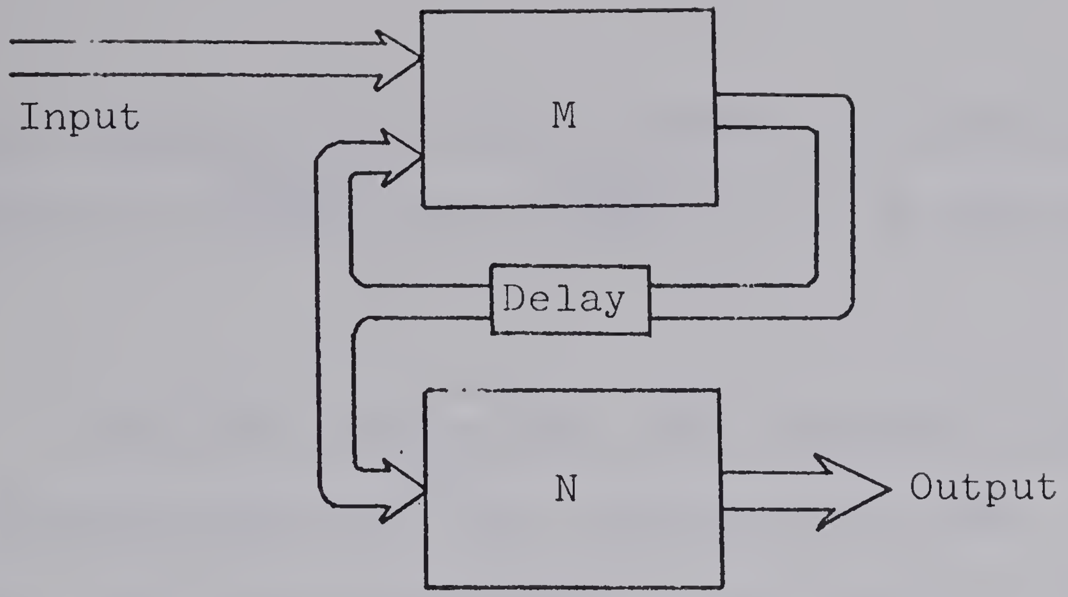


Fig. <4.1.1> A Moore Machine

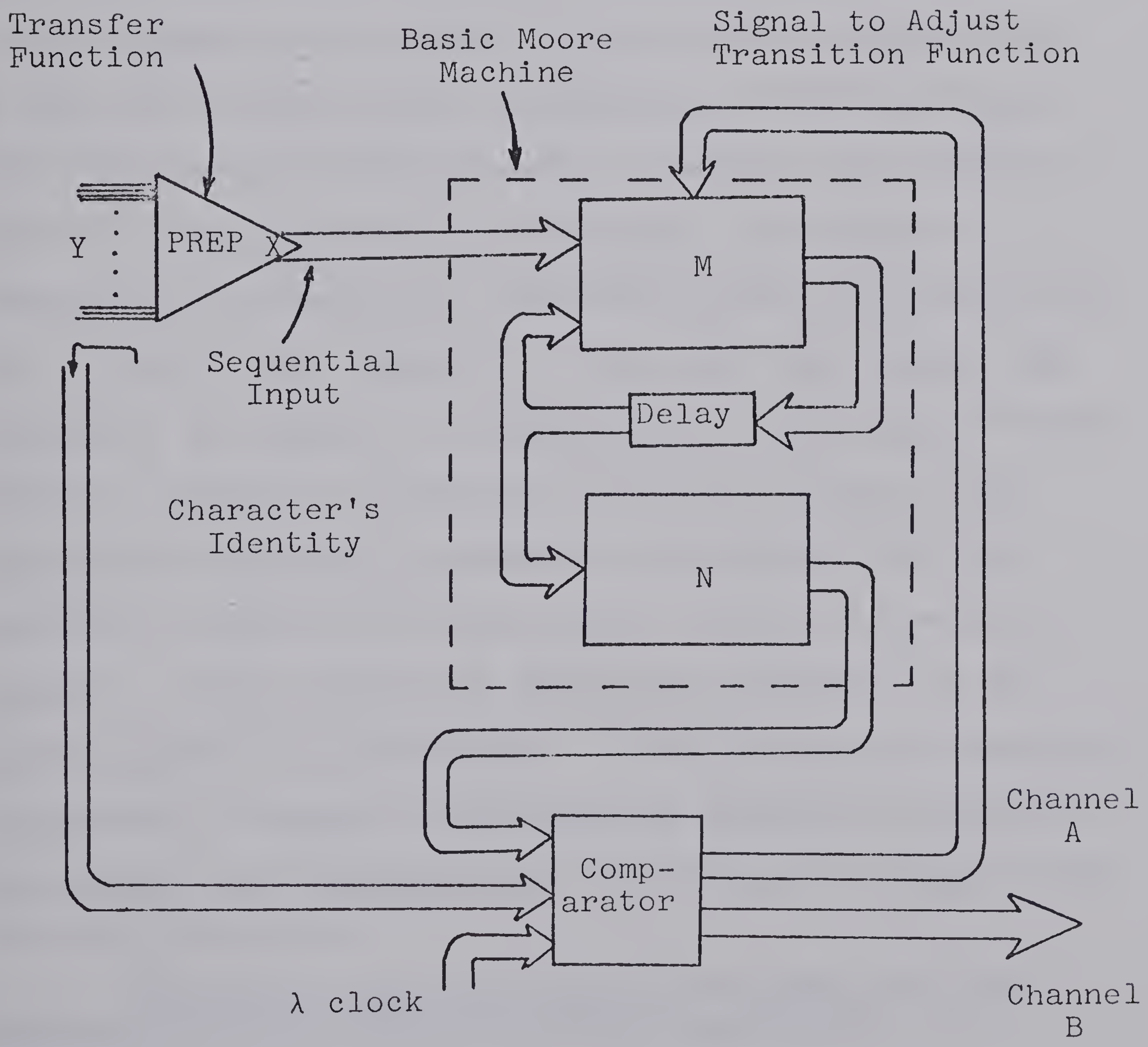


Fig. <4.1.2> The HPRS Machine

to-one, implying that for any pattern Y_1 differing from pattern Y_2 ($Y_1 \xrightarrow{PREP} X_1$ and $Y_2 \xrightarrow{PREP} X_2$), X_1 is not equal to X_2.

The HPRS categorizer, LEM, operates in two modes- the training mode and the classification mode. If the output from the preprocessor is introduced to LEM sequentially, the categorizer performs as a Moore machine with some extensions (see Fig. <4.1.2>). The next state of LEM depends on the present input and the present state. At the end of each pattern presentation (256 time steps)*, the output from the Moore machine specifies the category to which the input pattern is classified. This output information is passed to a comparator which is turned on by the λ clock at the completion of the 256 time steps. The comparator is composed of combinatorial logic which decides whether a pattern is classified correctly or not. The comparator's decision is based on information from the character's identity and the present output of the Moore machine. If the pattern is classified correctly, a new input sequence is initialized. If the pattern is classified incorrectly, a signal is delivered in Channel A (which is open during the training stage) to instruct a change in the

---

*256 time steps are required to input the 16x16 array which is the output from the preprocessor.

transition function, M, of the Moore machine.* The change in M should allow LEM to classify an identical character correctly at a future pattern presentation. Once the state transition function requires few-if any-changes, LEM has completed the training mode and is ready for classifying unknown patterns.

In the classification mode, the LEM is only concerned with identifying patterns. When the output from the model is transmitted to the comparator no decision is required. Rather the pattern's classification is sent out through Channel B of the comparator.

As a complete system, HPRS may be formally defined as a machine which composed of a transfer function and a Moore machine with extensions. With this basic machine definition completed, a more detailed and practical description of the HPRS model shall be devoted to the remainder of this chapter.

4.2 A Description of the Preprocessor

In this section the preprocessing system (PREP) used in the Hand Printing Recognition System (HPRS) is outlined.

*The ability to alter the transition function, M, is not a property of a Moore machine. For this reason the LEM model has been called a "Moore machine with extensions".

Sec. 4.2.1 examines some of the problems associated with the form of the character to be presented to PREP. Sec. 4.2.2 discusses a possible "hardware" implementation of PREP. Many problems were encountered during the design of the processing system and these will be mentioned in Sec. 4.2.2 as well. Sec. 4.2.3 is devoted to a presentation of the simulated model of PREP which is used in the experimentation.

4.2.1 The Form of the Character

A problem exists which is unique to the type of optical character recognition system designed here. PREP must be presented a transparent object or character. This is a drawback if hand-printed manuscripts are to be recognized. A solution is to photograph each page and extract the negative sheet of film containing transparent characters on a solid background [Holt (1968)]. This is an expensive method but requires no preprocessor time for it can be completed independent of the PREP system. Ideally, what is required is an inexpensive system which can take the light pattern of a character from a document and pass this pattern through a substance or special filter which reverses the light intensities. That is, the black print of the character becomes the illuminated portion and the intense light from the page becomes the darkened area.

The proper registration of a figure is important to the success of recognizing printed text. One helpful suggestion is to force the author of the text to write on a page that is spaced both vertically and horizontally much like standard Fortran coding sheets.

There are problems resulting from variations in thickness of the print, figure rotation and figure translation. For a writing instrument Munson [1968] found that a thin-lead mechanical pencil with HB lead gave the best performance. When the characters for the simulation program are drawn, the lines are coded relatively uniform in the thickness and approximately 1/8 the height of the character..(turn to Fig. <4.2.3.1>). Our present model does not make preprocessor adjustments to rotated and translated characters. Except for extreme cases, HPRS is capable of handling these variations.

4.2.2 A Hardware Implementation of PREP

Fig. <4.2.2.1> is a diagram which shall be referred to while describing the PREP hardware system. A coherent light (laser light) strikes the transparent object (in this case the capital letter A) and transmits the light pattern onto a beam splitter. The beam splitter allows some of the light to pass through and fall upon a spherical lens at a

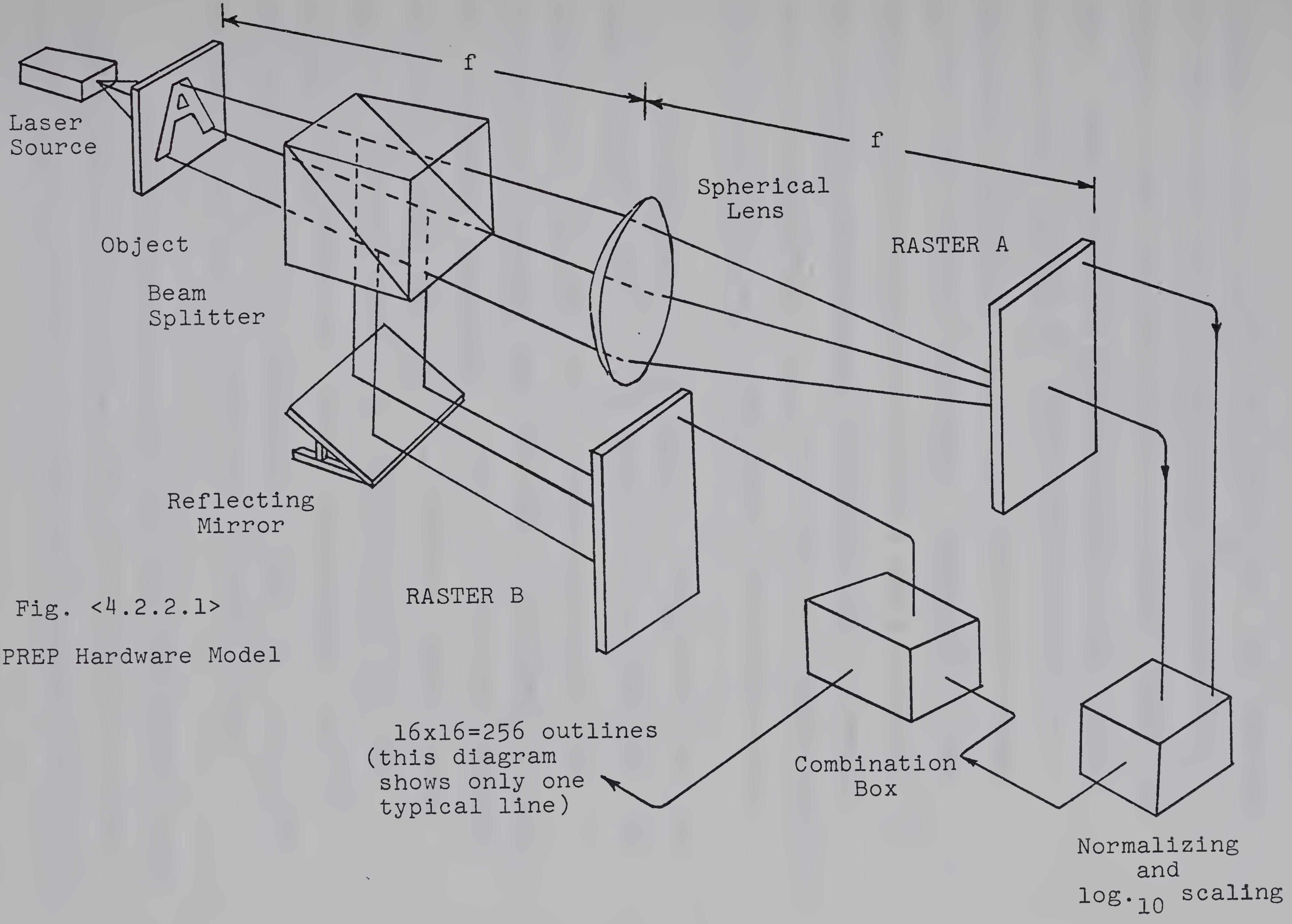

f
f
Laser Source
Object
Beam Splitter
Spherical Lens
RASTER A
Reflecting Mirror
RASTER B
16x16=256 outlines (this diagram shows only one typical line)
Combination Box
Normalizing and log.$_{10}$ scaling

Fig. <4.2.2.1>

PREP Hardware Model

distance f (focal length) from the object. The shape of the lens is such that an image of the transparency is produced on the transform plane, a distance f behind the lens. On the transform plane is placed a 16x16 raster of photocells (RASTER A). Each photocell sets up a resistance proportional to the amplitude of the light striking that point on the raster. Originally it was thought that the output from RASTER A would constitute all the necessary preprocessing required by HPRS. However, during the simulation it was found that the accuracy could be improved by making some refinements to the preprocessor.

As mentioned in the theory of Chapter III, the zero order frequency falls in the center of the transform. The intensity of the light in this middle portion of the transform is so bright that the value of information along the periphery is negliable. Thus, to record any frequency information about the object, a nonlinear damping scale must be used to decrease the sensitivity of the very intense centre area and enhance the peripheral sections. The logarithm_{10} of the magnitude of light is a convenient weighting scale which produces the necessary effect.

Another problem is that the light transmitted by an object varies with the area of the character of the object and its optical density. A large area character, such as M,

transmits more light than the character I and hence gives a stronger signal. To offset this effect, the output signals from the raster cells are scaled by a factor proportional to the zero frequency of each transform. This type of normalization can be "hardwired" and is cited as a requirement by Holeman [1968].

A third difficulty arises because the power spectrum at the filter (transform) plane captures only part of the information about the object. Whenever light diffracts, both its amplitude and phase are changed. Because photocells are only sensitive to the amplitude of the light, the simple system described thus far, yields no information about the relation of the various strokes or lines of the character. That is, it could not distinguish between a T, an L, or a + sign, each character having a horizontal and vertical bar. In an effort to overcome this problem, some spatial information must be extracted about the character as well as the frequency information from the optical transform. It was decided to focus the reflected light from the beam splitter-the light pattern of the character-onto a second 16x16 raster of photocells (RASTER B). The output from RASTER B is then combined with that of RASTER A to form a total output for the PREP system. It was found in this research that by weighting the information from RASTER A and RASTER B equally, the best results were obtained.

However for a particular pattern recognition problem the output from RASTER B need only be weighted enough to alleviate the problem of losing the phase information.

The introduction of RASTER B also eliminates another problem that is inherent in the optical transform. The difficulty arises from the fact that the transforms are invariant under 180° rotation of the character. The letters M and W, and A and V have very similar transforms although they appear very different as characters.

The hardwired connections between RASTERS A and B are one for one. The photocell in row x and column y of RASTER A connects with the photocell of row x and column y of RASTER B. The weighting of RASTER A and RASTER B photocells may be altered by changing the load resisters in the respective circuits. A 16x16 wire matrix will carry the output signal from PREP. All signals are made to conform with a grey scale of 0-9. The output from PREP now feeds into the LEarning Machine (LEM) which will be discussed in Sec. 4.4.

4.2.3 The Simulated Model of PREP

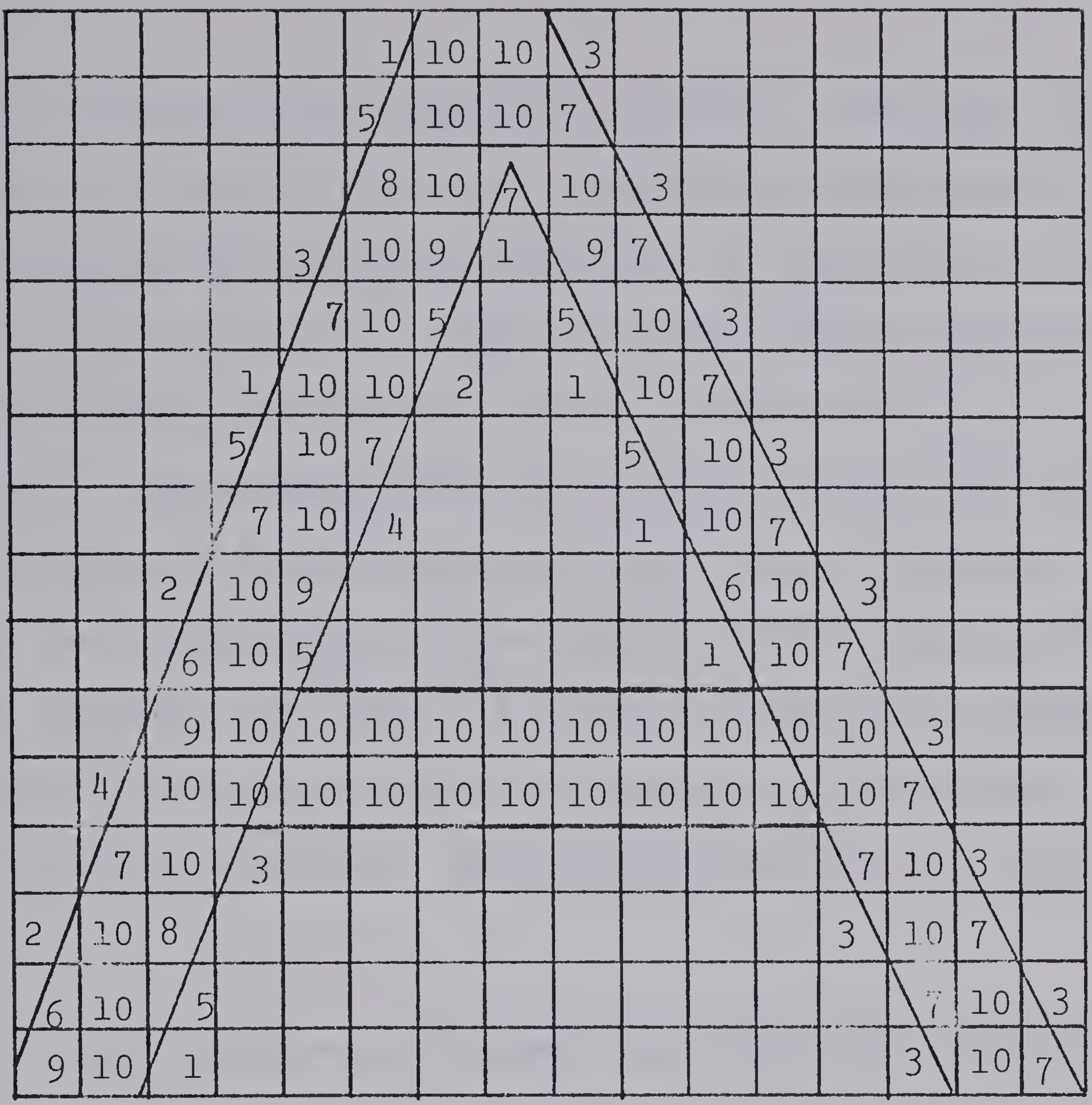
The program which simulates PREP is written in the FORTRAN IV language, runs on a 360/67 computer under an OS/MVT environment, and uses the level H compiler. The

program reads in a 16x16 array, M, which is a representative of the pattern of light as it passes through the transparent object (character). Fig. <4.2.3.1> demonstrates the coding necessary to represent the letter A.* Many other simulations use a binary matrix to describe an object [Munson (1968], but it is felt that the nonbinary type of coding is more representative of the relative intensity of the light from the object as it would strike RASTER B. As well, nonbinary coding aids in the processing of a more accurate, discrete optical transform of the character.

To simulate the optical transformation of the light as it passes through the spherical lens, the Fast Fourier Transform (FFT) program as written by G. Sande [1968], is used. An excellent description of the algorithm may be found in [G-AE Subcommittee on Measurement Concepts (1967)]. It is mentioned in the theory of Chapter III that the FFT procedure must be applied twice (first along the rows and then along the columns of the matrix M) to accommodate the two dimensional case. The result of the procedure, however; is a shifted transform and not the true transform as viewed

---

*Some of the characters tested in this research were from data collected originally by Munson. Since Munson's characters were binary encoded, it was necessary to write a computer program to convert them to a nonbinary representation so that they could be used in this research. A description of the conversion routine shall be given in Chapter V.



					1	10	10	3							
					5	10	10	7							
					8	10	7	10	3						
				3	10	9	1	9	7						
				7	10	5		5	10	3					
			1	10	10	2		1	10	7					
			5	10	7				5	10	3				
			7	10	4				1	10	7				
		2	10	9						6	10	3			
		6	10	5						1	10	7			
		9	10	10	10	10	10	10	10	10	10	10	3		
	4	10	10	10	10	10	10	10	10	10	10	10	7		
	7	10	3									7	10	3	
2	10	8										3	10	7	
6	10	5											7	10	3
9	10	1											3	10	7

Fig. <4.2.3.1> A Hand-Coded Letter

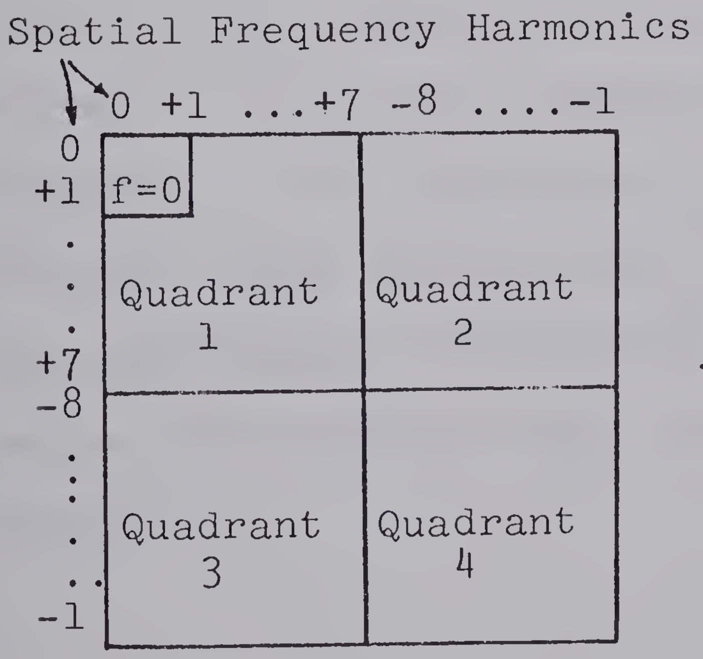


Fig. <4.2.3.2>
Shifted Transform from
Discrete F.F.T. Algorithm

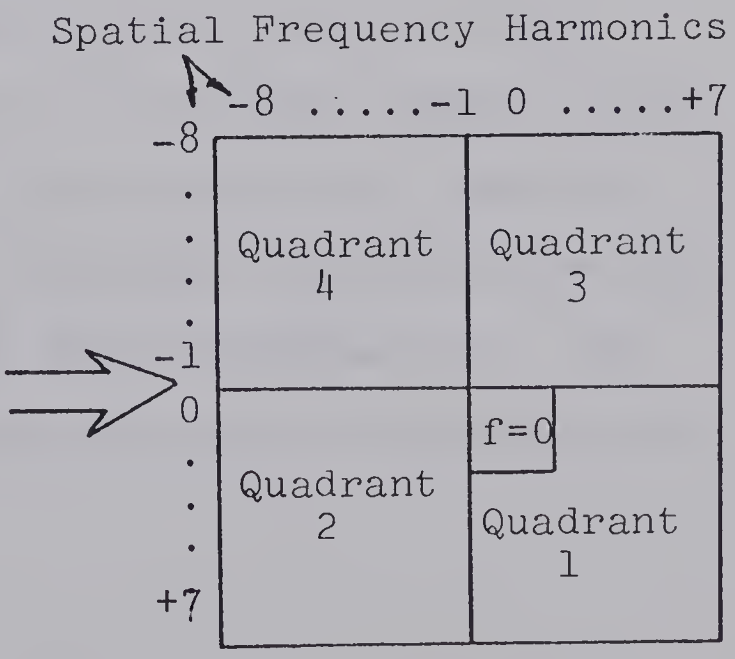


Fig. <4.2.3.3>
Unshifted Transform-
Actual Optical Transform

at the transform plane [Barkdoll (1968)]. See Fig. <4.2.3.2>. The discrete case of the true transform is achieved by interchanging the diagonal quadrants of the matrix to the form in Fig. <4.2.3.3>. Hence the zero order frequency of the transform is now in the center of the matrix as is required. The normalizing and scaling features are simulated by dividing the transform by the zero order frequency and taking the logarithm_{10} of the result. Then a grey scale of 0-9 is applied such that the largest element is assigned a value of nine and the smallest element a value of zero. Fig. <4.2.3.4> shows the grey scale transform of the letter A.

The transform and object matrices which represent the output from RASTER A and RASTER B, respectively, are weighed by an alpha factor and combined to form the output from the simulated PREP program. An alpha factor of .8 results in the elements of the transform matrix being weighed by .8 and the elements of the object matrix being weighed by 1-.8=.2 before the two matrices are combined. Different alpha factors yield different learning rates and test performances on identical sets of characters. The results for different alpha factors shall be discussed in Chapter V.

0	0	0	1	0	1	1	1	0	1	1	1	0	1	0	0
0	0	0	1	0	0	2	2	0	2	2	0	0	1	0	0
0	0	0	0	0	0	2	3	1	2	2	0	0	0	0	0
0	0	0	1	1	0	0	2	3	0	0	0	0	1	0	0
0	0	0	2	1	0	2	3	4	3	2	0	1	1	0	0
0	0	0	2	0	1	4	5	3	5	4	1	1	1	0	0
0	3	5	5	3	0	5	5	5	6	5	0	4	6	5	3
0	0	2	5	7	8	8	5	5	6	8	8	7	4	2	1
0	0	0	2	0	4	6	9	0	9	6	4	0	2	0	0
0	1	2	4	7	8	8	6	5	5	8	8	7	5	2	0
0	3	5	6	4	0	5	6	5	5	5	0	3	5	5	3
0	0	0	1	1	1	4	5	3	5	4	1	0	2	0	0
0	0	0	1	1	0	2	3	4	3	2	0	1	2	0	0
0	0	0	1	0	0	0	0	3	2	0	0	1	1	0	0
0	0	0	0	0	0	2	2	1	3	2	0	0	0	0	0
0	0	0	1	0	0	2	2	0	2	2	0	0	1	0	0

Fig. <4.2.3.4>

Grey Scale Transform of the Letter A.

4.3 A Description of the Categorizer

Sec. 4.3 outlines a possible "hardware" model of LEM, the HPRS categorizer, and describes the program used to simulate this model. The ideas for the hardware are derived from the Learning Matrix theory as discussed by Steinbuch and Piske [1963], Steinbuch [1965], and Kazmieczak and Steinbuch [1963]. Some preliminary notation for this theory is introduced in Sec. 4.3.1. Sec. 4.3.2 is devoted to a description of a possible hardware configuration. In Sec. 4.3.3 a brief explanation of the simulation model of LEM is presented. Some mention is made of the problems encountered in programming the model.

4.3.1 Learning Matrix Notation

The learning matrix is a matrix-like circuit structure of rows and columns. The intersection points of the matrix are formed by connecting elements (eg. transfluxors, wound ribbon cores, electrochemical cells, magnetic film cores). The learning matrix functions in three modes: a learning mode, an EB (classifying) mode and a BE (readout) mode.

During the learning mode, the characteristics of an object or pattern are presented to the columns as

nonbinary* signals via the preprocessor transducers. Simultaneously a meaning of a subcategory to be associated with these sets of pattern characteristics is applied in the form of a signal to one of the rows. During the learning mode the connective elements form conditioned connections (i.e. the conductance of the element becomes proportional to the current passing through the connections). Thus, when a signal is applied to a designated row, the conductance of each element along that row is changed to represent the characteristics of the input pattern signal. By applying the training algorithm (as described in Chapter III), the conductance of a row of connective elements may be made to represent the prototype vector for a subcategory.

During the EB mode (from the characteristics - Eigenschaften to the meaning - Bedeutung), the nonbinary encoded characteristics of an input pattern are presented to the columns of the learning matrix. If a row relay is closed during the input presentation, a current is set up along the row wire proportional to the dot product of the prototype vector (the conductances, G_{kj}, of the connecting elements), and the tranducer voltage, X_j, from the column wires. Therefore,

---

*The characteristics may also be binary encoded However, for the type of machine under consideration this is not practical.

$$\text{current in row } k = \sum_{j=1}^{m} X_j \cdot G_{kj} \qquad \ldots \; (4.3.1.1)$$

where m is the number of characteristics in any input pattern.

During the BE mode (from the meaning-Bedeutung-to the characteristics-Eigenshaften), a signal is applied at a designated row so that a current is set up in the matrix column. The current created in the columns is a read-out of the characteristics of that row (or the prototype vector for that subcategory). The BE mode of operation is particularly useful for finding the squared component terms used in minimum-distance classifications. As well, the EB and BE mode may be combined to find the dot product between two different rows (or subcategories) by applying a read-out signal at one row and closing the relay on the other row.

4.3.2 A Hardware Implementation of LEM

To train the categorizer using the algorithm described in Chapter III, two learning matrices are required. Both matrices must be capable of operating in all three modes.

When creating new subcategories or altering present ones, the machine (see Fig. <4.3.2.1>) must operate in the learning mode. At the beginning of the learning mode the

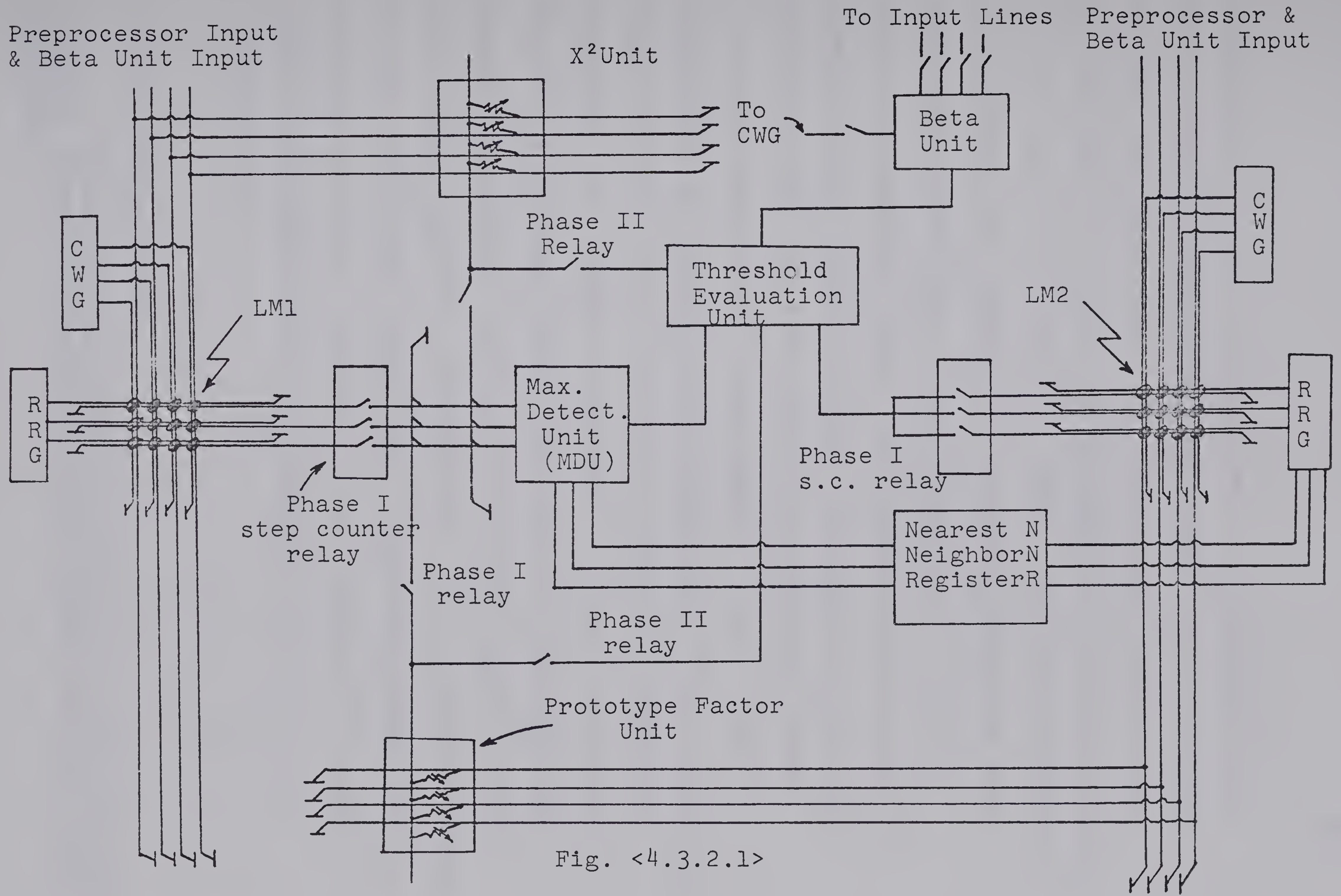


Fig. <4.3.2.1>

A LEM Hardware Model

transfluxors* of the selected row (a row representing a new or existing subcategory) are brought into a defined initial, magnetic, state. An input signal is sent from the Beta Unit (the unit used to calculate β in Eq.(3.2.4.2)) to the column write generator (CWG). The CWG furnishes the column wires with impulses of increasing magnitude which effect the connections of the selected row. In this manner the magnetic state of the transfluxors are made to change in small quantized steps. By means of a row read generator (RRG) and with the aid of a sensing current, a voltage proportional to the magnetic state of the transfluxor is induced in the column wire of the transfluxor's small hole (see. Fig. <4.3.2.2>). After the current is amplified, its voltage is applied to a signal comparator. When the difference between the read-out signal and the signal initiated by the Beta Unit is less than a certain error limit, the CWG is switched off. The process of forming the new conditioned connections in the transfluxors for one training pattern is completed. The formation of the new conditioned connections also designates the end of the learning mode.

---

*It was decided to use transfluxors as the connecting elements in the configuration of the hardware machine being described.

The categorizer decides if a new subcategory is to be formed or a present subcategory is to be adjusted on the basis of the results of classifying a training pattern. To classify a pattern (whether it be from the training set or testing set), the machine employs the other two modes of operations, the EB and BE modes, in two distinct phases. In Phase I, the minimum distance subcategory classification is found for the input pattern. In Phase II, the β-factor is calculated so later, in the learning mode, it may be used to complete the necessary adjustments to the learning matrix connections.

To find the minimum Euclidean distance category, LM1 (learning matrix one) is placed in the EB mode. In this mode the dot product $(X.P_j^{(i)})$ of any designated subcategory and the input pattern may be calculated. The X^2 term is computed in the X^2 unit by generating a current that is proportional to minus one half the sum of the squares of input voltage from the preprocessor transducers $(1/2X^2)$. Simultaneously, in LM2, the BE mode is enabled. With the BE mode on, the $-1/2P_j^{(i)^2}$ term may be computed for any designated subcategory. This is done by reading out the corresponding row of the designated subcategory from LM2 so that the current produced, flows to the PROTOTYPE FACTOR Unit. This unit (as does the X^2 unit) uses rheostats to generate a current that is proportional to minus one half the sum of the squares of the voltage produced from the LM2 read-out.

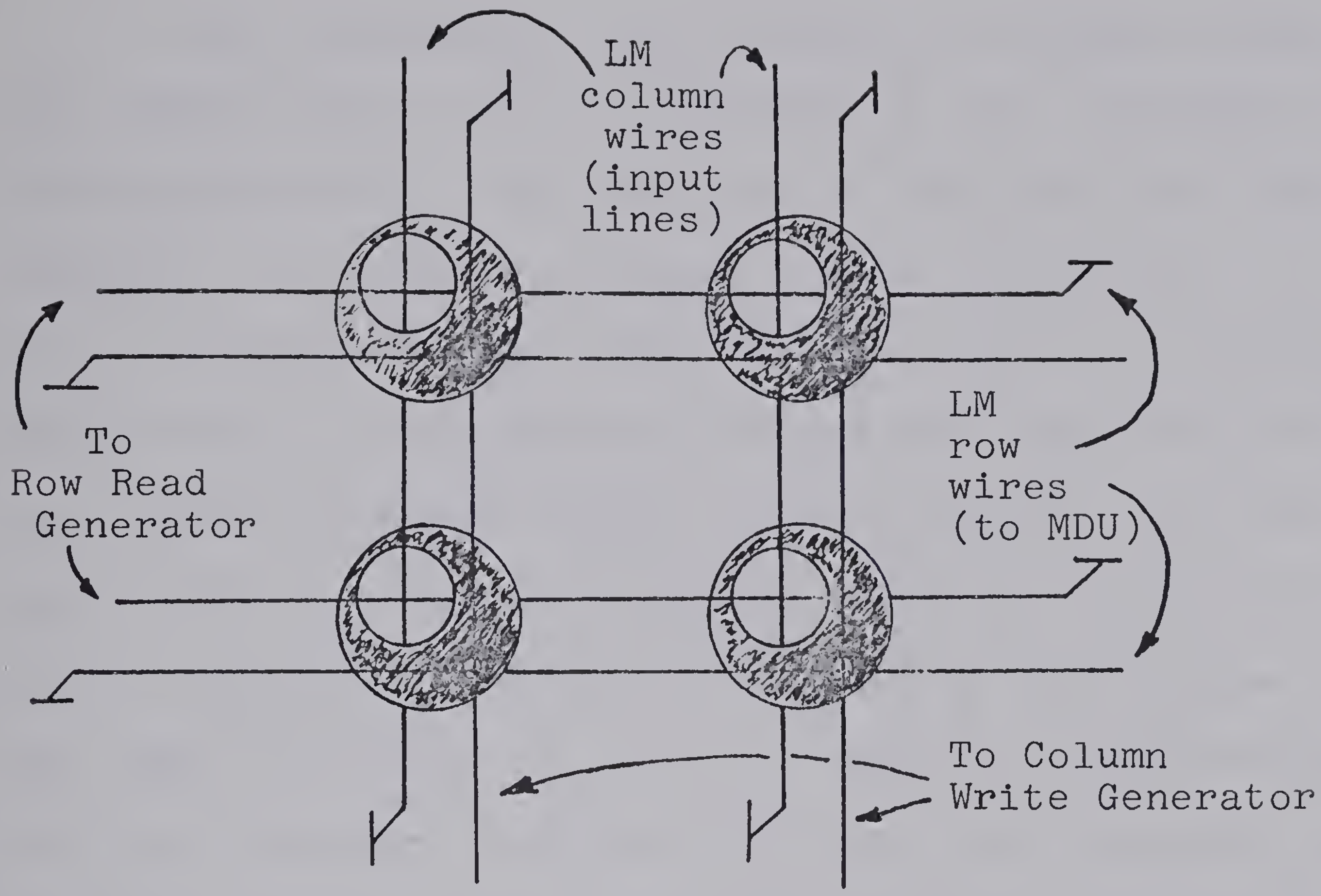


Fig. <4.3.2.2>

Diagram of a nonbinary 2x2 learning matrix with transfluxors.

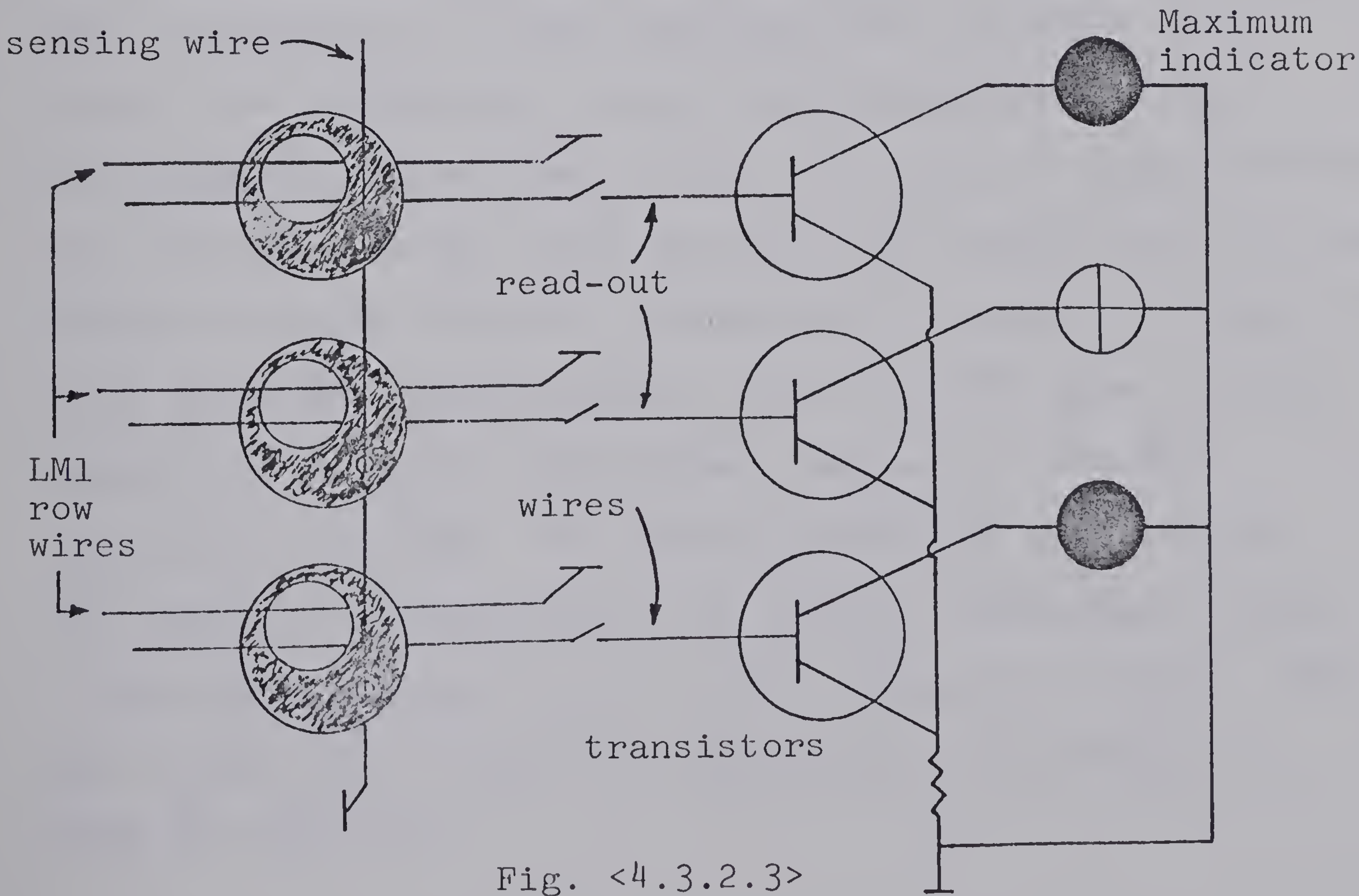


Fig. <4.3.2.3>

Maximum Detector Unit for three subcategory case.

The "designated" row is chosen by using a Phase I step counter relay which is attached to the row wires of both learning matrices. The step counter delivers a top down sequence (i.e. first row-first, second row-second, etc.). Hence, the discriminant value for each row of the learning matrices may be estimated by hooking the lines from the X^2 Unit and the Prototype Factor Unit in parallel with row wires of LM1. These discriminant values are stored in a set of transfluxors which are situated in the Maximum Detector Unit (MDU) (see Fig. <4.3.2.3>). After all discriminants have been computed the EB and BE modes are terminated and the step counter reset. A current is produced in the MDU transfluxors inducing a nondestructive read-out of the discriminant values for all subcategories. These values are feed, in parallel, to the transistorized circuitry which detects the maximum row. Once the maximum row is found the categorizer must check to see if a correct classification has resulted. If the input pattern does indeed belong to the maximum category, then the categorizer is ready for Phase II. If an incorrect classification occurs the MDU gate for the maximum row is opened and another read-out of the MDU transfluxors is made. The second maximum is then tested. This process continues until the largest discriminant value of the subcategories of the correct category is found. Once such a row (call it row^r) is discovered, the category is ready for Phase II.

In Phase II the categorizer must decide if the input pattern lies within the threshold value for the subcategory represented by row r. To calculate the threshold as given by Eq.(3.2.2.0), the row of the nearest neighbor subcategory to row r must be located. To complete this task, a signal is sent to the Nearest Neighbor Register (NNR). Once the row corresponding to the nearest neighbor subcategory is found in the NNR, the LM2 read-out generator is turned on at the nearest neighbor row. At the same time all the Phase I step counter relays are set to row r. By placing LM1 in BE mode, the term $-1/2P_i^{(j)^2}$ is computed in the X^2 unit (assuming subcategory $C_i^{(j)}$ corresponds to row r). By placing LM2 in BE and EB mode simultaneously, the dot product $P_i^{(j)}.P_k^{(m)}$ and the term $-1/2P_k^{(m)^2}$ of the threshold value are calculated (since the nearest neighbor category $C_k^{(m)}$ is read-out of LM2). The $-1/2P_k^{(m)^2}$ term is computed in the Prototype Factor Unit. These three values are combined at the Threshold Evaluating Unit. The resulting threshold value is then sent to the Beta Unit where it is compared to the discriminant value of row r. If adjustments are necessary (the criteria and the adjustment formula are given in Chapter III), a new β-factor is calculated and the categorizer returns to the learning mode. At the completion of the learning mode, a read-out of each row is made in LM1. By using the MDU, the maximum discriminant values for

each read-out may be found. The row corresponding to this value is then stored in the appropriate Nearest Neighbor Register. In this manner the system is continually updated.

Once the categorizer has identified the training set, it is ready to classify patterns from the environment. If the categorizer employs Phase I of the training classification procedure described above, an Absolute MED classification results. If a Relative Reliability classification is desired, the categorizer must execute Phase II of the training classification procedure as well as Phase I.

While many details of the design of the categorizer are omitted, enough information is present to visualize the implementation of a hardware machine. In the next section a simulation of the hardware model is considered. The software model consists of a program written for a general purpose computer to simulate the action of the hardware device.

4.3.3 The Simulated Model of LEM

The program which simulates LEM is written in PL1 programming language, runs on a IBM 360/67 computer under an OS/MVT environment and uses the level F compiler. The control of the simulated machine is embodied in an iterative loop situated in the main procedure of the program (see

Fig. <4.3.3.1>). Each input pattern, whether from the training set or the test environment is passed through this loop. Within the loop, procedures are called which simulate the different modes of the hardware machine. The three most significant procedures are MAXDET, LEARNING and STUDENT. MAXDET simulates the EB and BE modes of LEM and is responsible for all pattern classifications. LEARNING decides what type of learning is to take place (mode seeking or error-correcting), computes the β-factor and adjusts the prototype vectors (conditioned corrections) as described when discussing the learning mode of the hardware model. STUDENT (see Fig. <4.3.3.2>) locates the closest, correct, minimum-distance subcategory and decides whether a new subcategory is to be formed or not.

The simulation program of the categorizer is a surface-fitting learning model. The 16x16 input array from the preprocessor is a set of discrete points removed from a surface in three-dimensional Euclidean space. A point-by-point correlation (autocorrelation of degree 0) is made between the input pattern surfaces and the set of prototype surfaces which represent subcategories. Surface fitting models have many useful applications in the general field of pattern recognition. Some of these applications shall be mentioned in Chapter VI.

Problems with storage and time limits are encountered

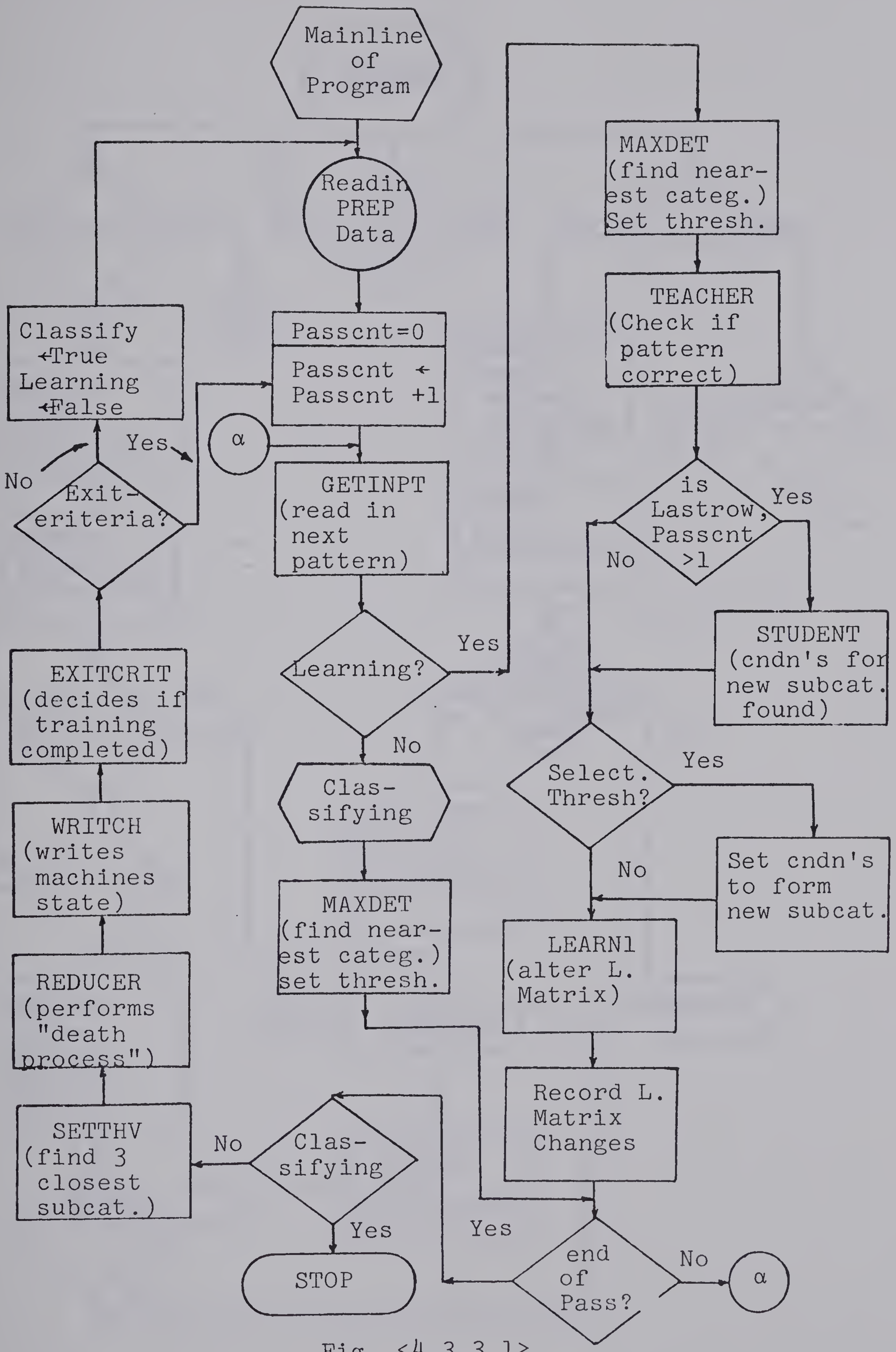

Mainline of Program
Readin PREP Data
Passcnt=0
Passcnt ← Passcnt +1
α
GETINPT (read in next pattern)
Learning?
Yes
No
Clas-sifying
MAXDET (find near-est categ.) set thresh.
Classify ←True Learning ←False
Yes
No
Exit-eriteria?
EXITCRIT (decides if training completed)
WRITCH (writes machines state)
REDUCER (performs "death process")
SETTHV (find 3 closest subcat.)
No
Clas-sifying
Yes
STOP
MAXDET (find near-est categ.) Set thresh.
TEACHER (Check if pattern correct)
is Lastrow, Passcnt >1
Yes
No
STUDENT (cndn's for new subcat. found)
Select. Thresh?
Yes
No
Set cndn's to form new subcat.
LEARN1 (alter L. Matrix)
Record L. Matrix Changes
end of Pass?
Yes
No
α

Fig. <4.3.3.1>

The LEM Program

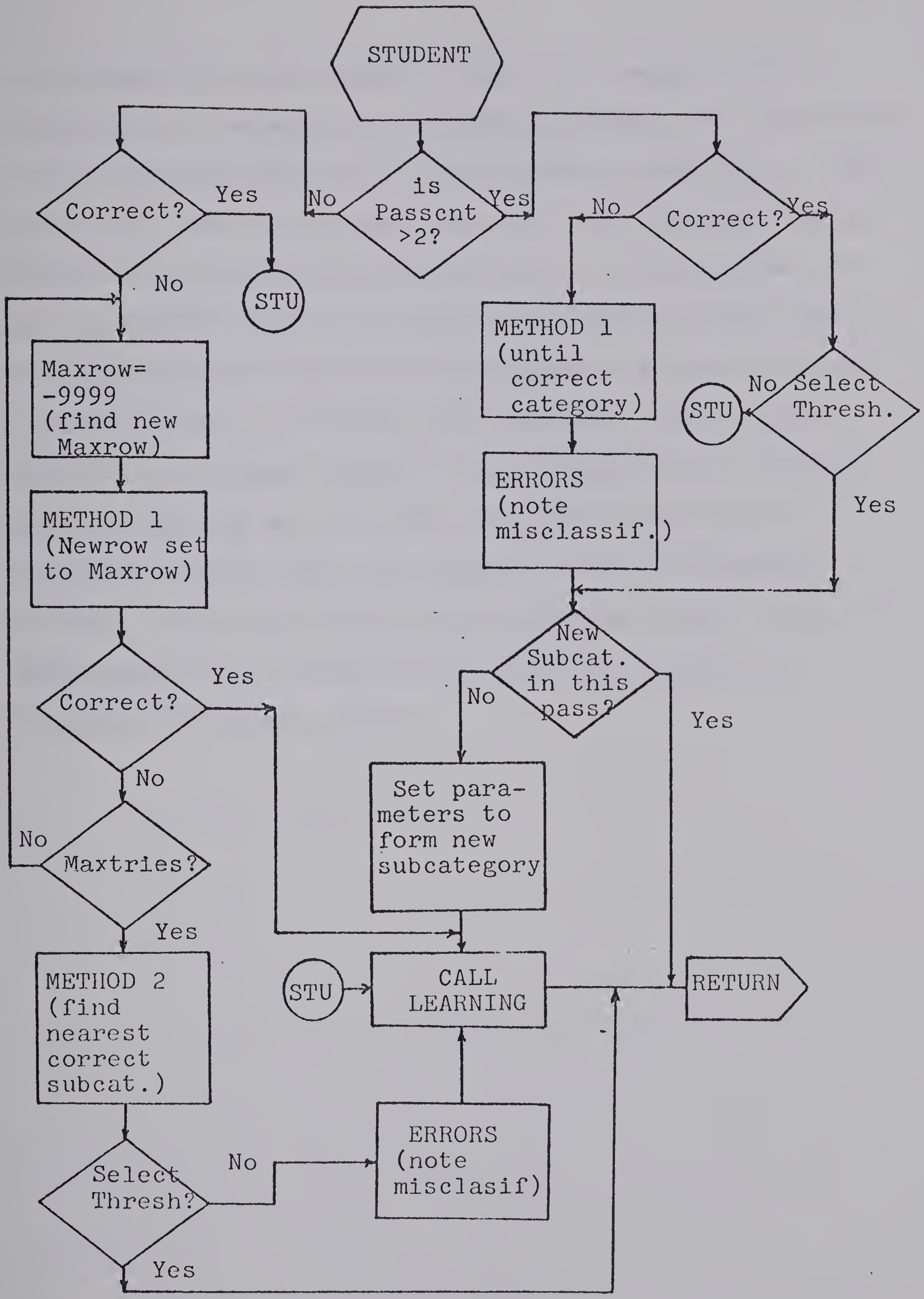

STUDENT
is Passcnt >2?
No
Yes
Correct?
Yes
No
STU
Maxrow= -9999 (find new Maxrow)
METHOD 1 (Newrow set to Maxrow)
Correct?
Yes
No
Maxtries?
No
Yes
METHOD 2 (find nearest correct subcat.)
Select Thresh?
No
Yes
ERRORS (note misclasif)
STU
CALL LEARNING
RETURN
Correct?
No
Yes
METHOD 1 (until correct category)
ERRORS (note misclassif.)
Select Thresh.
No
STU
Yes
New Subcat. in this pass?
No
Yes
Set parameters to form new subcategory

Fig. <4.3.3.2>

The Student Program

in implementing the program. Under the present system installation (release 16 of OS/MVT), programs are run within a core storage partition of approximately 200k bytes. This partition limit is attained when over 300 subcategories are created and stored in the categorizer. A learning set of 230 characters requires seventy-five minutes of CPU time to be appropriately trained by the program. A total of five runs are needed to complete the training. At the end of each run the present state of the categorizer is outputed. At the beginning of the follow run, the categorizer is returned to state that was outputed before continuing further. After the machine has reached a trained state, approximately 4.5 seconds of CPU time is required to classify an unknown pattern.

CHAPTER V

EXPERIMENTAL RESULTS

Initial testing of the HPRS model was performed on a small set of hand-coded characters created by the researcher. The limited results of this testing are described in Sec. 5.1. Extensive testing was completed on over one thousand hand-printed characters from Munson's Multiple-Coder File. For a comparison, the results achieved by the Munson machine and HPRS are presented under section heading "Extensive Testing" (Sec. 5.2). Sec. 5.3 is a discussion of the performance of the HPRS model.

5.1 Initial Testing

There were two purposes for the initial testing of HPRS with hand-coded characters.

1) To insure the HPRS simulation program ran correctly before going on to extensive testing.

2) To analyze the performance of HPRS using different α-weighting factors (as mentioned in Sec. 4.2.2). From this analysis some judgement could be made of what α-factor should be used for larger test runs.

The initial test set consisted only of the letters

(A-Z) of the alphabet. In an effort to reduce computer time and lines of output for each run, the twenty-six different pattern classes were grouped into three sets (<A,F,H,L,T,X,Y,V><B,C,D,E,O,P,R,U><G,I,J,M,N,Q,S,W,Z>).[*] After the program was found to run correctly, it was trained and tested on some typical hand-coded, hand-printed characters. The results of applying 16, well-formed test patterns (A,B,C,D,E,F,G,I,L,M,N,P,R,W,X,Y) are listed below.

Table <5.2.1> Results of Initial Testing

				Type of Classification				
[**]	training	# of	# of subcat.	MED		Rel. Reliability		
α	Set size	classes	formed	# cor	# wrg.	# cor	# rej.	# wrg
1.0	82	26	56	9	7	2	12	2
.8	"	"	53	12	4	5	9	2
.7	"	"	48	10	6	8	5	3
.6	"	"	46	10	6	7	7	2
.5	"	"	46	13	3	10	5	1
.4	"	"	42	12	4	9	4	3
.0	"	"	42	11	5	9	5	2

Legend: cor = correct, wrg = wrong, rej = rejected.

---

[*] The three sets were formed on the basis of the Euclidean Distance measure between each class of a special test run. In the special test run, each of the twenty-six letters was represented by a prototype pattern which was free of noise (TYPE ONE and TYPE TWO). Those patterns that were closest together in Euclidean space were taken to be very similar and placed into a common set.

[**] An alpha value of 1.0 means that RASTER A is weighted 100%.

Because of time considerations, extensive testing could not be completed for a variety of α weightings. Therefore only preprocessor data with an α-factor of .5 (the weighting factor that was most accurate during initial testing) was thoroughly examined.*

5.2 Extensive Testing

Hand-coded characters could not be used to adequately test HPRS for two reasons:

1) Since 10-15 minutes are required to hand-code a single character, it would have taken between 150 to 200 hours to code a large enough sample (say 1000 characters) for a proper investigation of the simulated model.

2) Even if a large sample could be hand-coded the characters created might not be representative of characters from an authentic hand-printed document.

For these two reasons, a large realistic set of characters was needed to test the simulation model. Munson's Multiple-Coder File fulfilled this need. Printed data from 49 individuals was included in the Multiple-Coder file. Each person was asked to print several 46-character alphabets (10 numerials, the 26 uppercase letters, and the special symbols [=*/+-.,$]) on a coding sheet. The first 3 alphabets from each sheet were taken for the file. The coders were

----

*Since the initial testing was carried out on a limited set of well-formed characters, the α-factor of .5 should be construed as a convenient but not necessarily optimal value for extensive testing.

asked to print naturally, being neither especially casual nor especially meticulous. Five human subjects were asked to classify the characters in 17 of the test alphabets. The error rates ranged from 3.0% to 6.4%, with an average of 4.5%. A plurality vote among the five responses yielded an error rate of 3.2%.

5.2.1 Munson's Results

Munson's training set was composed of the alphabet from the first 32 persons (96 alphabets, 4416 characters). The data from the remaining 17 persons was used for testing (51 alphabets, 2346 characters). A categorizer, CALM, of the PWL type (Nilsson [1965]) was used to classify the characters on the basis of the feature vectors generated by the two preprocessors, PREP and TOPO. Munson conducted his experiments on the Multiple-Coder File using three conditions.

In Condition I, the characters were preprocessed by PREP in all nine views*. It required nine iterations for the categorizer to encounter each view of each pattern. The nine response for each category were added together to form

*Eight additional views were captured by translating the character two positions in each of the vertical and horizontal directions.

cumulative responses which were used as the basis for classification. A PWL learning machine with a maximum of two subcategories per category were used in Condition 1.

In Condition 2, the characters were preprocessed by TOPO. Owing to computer restrictions (which Munson did not elaborate upon), only one subcategory per category was formed by the learning machine - that is, the categorizer performed as a linear classifier rather than a PWL classifier.

In Condition 3, the responses of the learning machine in Conditions 1 and 2 were added together and taken as a new basis for classification of each pattern tested. The combined system was tried in an attempt to improve the classification performance by using the information from two different preprocessor measurement domains.

The results of Munson's experiments are presented below:

Table <5.2.2> Munson's Results

Cond.	Preprocessor	# of Iterations	Training Patterns	Test Patterns
1	PREP, 9 views	18	65%	78%
2	TOPO	4	84%	77%
3	combined	-	-	85%

Munson did not reveal information about the size of his simulation programs PREP, TOPO and CALM - or how much time was required to process and classify a character. The single comment he did make was that "it sometimes took days to accomplish the experimental runs".

5.2.2 HPRS Refinements for Extensive Testing

Before presenting the results of the extensive testing, some mention should be made of two difficulties that arise in Munson's Multiple-Coder File as applied to the HPRS model. First, the binary character representation (see Fig. <5.2.2.1>) is not compatiable with the PREP preprocessor which requires the use of a nonbinary scale. To produce a proper nonbinary input a quantizing technique is performed on the 24x24 binary raster which represents the character. In the quantizing procedure overlapping (3x3) submatrices of the binary matrix are summed to yield nonbinary (0-9) values. Fig. <5.2.2.2> illustrates the results of quantizing the binary character in Fig. <5.2.2.1>. Close examination shows that this quantizing technique produces a very good nonbinary representation of the original character.

A second difficulty that must be overcome, results from the different scale employed by Munson in collecting the Multiple-Coder File than the scale that is used in the

```
X X X X X X X X X X X X X X X X X X X X X X X X
X                                             X
X                                             X
X                                             X
X                 1 1 1 1 1 1 1 1 1 1 1       X
X             1 1 1 1 1 1 1 1 1 1 1 1 1       X
X               1 1             1 1 1 1       X
X                             1 1 1 1 1       X
X                           1 1 1 1 1         X
X                       1 1 1 1 1             X
X                   1 1 1 1 1                 X
X                 1 1 1 1                     X
X               1 1 1 1                       X
X             1 1 1 1                         X
X           1 1 1                             X
X         1 1 1 1             1               X
X         1 1 1 1 1 1 1 1 1 1 1 1 1           X
X           1 1 1 1 1 1 1 1 1 1 1 1           X
X             1 1 1 1 1 1 1 1 1               X
X                                             X
X                                             X
X                                             X
X                                             X
X                                             X
X X X X X X X X X X X X X X X X X X X X X X X X
```

Fig. <5.2.2.1>. Munson's Binary Coded Figure

```
X X X X X X X X X X X X X X X X
X         1 1 2 1 2 1 1       X
X     1 4 5 6 6 6 6 6 5 2     X
X     2 5 6 4 5 5 8 9 8 3     X
X       2 1     3 7 9 6 2     X
X         1 2 5 6 7 5 3       X
X       1 4 7 7 5 3 1         X
X     1 4 8 7 4 1             X
X   1 4 7 6 3                 X
X 1 4 8 7 4 2 2 2 3 1         X
X 1 5 8 8 6 6 6 7 7 4 1       X
X   2 6 7 8 7 8 7 6 3 1       X
X     1 3 3 3 3 3 1           X
X                             X
X                             X
X X X X X X X X X X X X X X X X
```

Fig. <5.2.2.2>. Nonbinary Representation of Binary Figure

initial testing of the hand-coded characters. Because a pattern from the Multiple-Coder File is centered and is small when compared to the full (24x24) mask (see Fig. <5.2.2.1>), the peripheral sections of the mask are never "ON" as they are in the normalized hand-coded character. For this reason it was decided to combine information from the optical transform with the outer sections of the light pattern formed on RASTER B. That is, RASTER A (containing the optical transform information) and RASTER B would be wired so that the lower order frequency information of the transform would fall along the periphery, and the light pattern of the character would lie in the middle of the combined PREP output. Thus, an effort has been made to separate the information from each raster as much as possible. The greater the overlapping of the two sources of information, the greater is the possibility of ambiguous representation of the character.

The changes brought about by the two difficulties just discussed, are necessary only to make HPRS compatible with Multiple-Code File data and do not deter from the basic model. However, because of the large training set, two changes were made to the basic model. In an effort to reduce large numbers of maverick subcategories, those subcategories containing a single pattern for two consecutive iterations of the training set were eliminated. At the

completion of the training period a final death procedure was implemented in which all subcategories representing one pattern were destroyed. Those two additional death processes aided in reducing the number of subcategories which is extremely important when considering LEM's hardware costs and time to process a character.

5.2.3 HPRS Results

In the extensive testing of HPRS, a LEARNING SET of 460 characters (ten alphabets-two alphabets from each of five different subjects) was chosen. The training period was terminated after six passes of the learning set at which time LEM could identify 86.7% of the characters correctly. The model was then tested on two different sets of characters, a KNOWN SET and an UNKNOWN SET. The known set consists of 230 characters (five alphabets-one from each of the subjects that created the learning set). The unknown set is made up of 1150 characters (25 alphabets from 9 subjects). The subjects who created the learning set were not among those who printed the unknown set. The test results for each member of the 46 character alphabet are shown in Tbl. <5.2.3.1> and Tbl. <5.2.3.2>.

5.3 Machine Performance

In the discussion of the machine performance each

	# of	Training Set			Known Set			Unknown Set			
	Sub-	MED	RR		MED	RR		MED	RR		
Cat	cat.	#cor	#cor	#rej	#cor	#cor	#rej	%cor	%cor	%rej	MOMC
1	3	7	5	2	2	2	2	64	44	36	/
2	3	8	6	3	3	1	4	44	8	76	I
3	2	9	3	7	4	1	3	44	24	64	2
4	3	6	5	4	3	2	3	36	20	68	A
5	3	9	7	3	5	4	1	68	28	64	S
6	2	9	9	-	2	2	3	68	36	60	G
7	3	10	9	1	4	1	4	84	68	28	J
8	4	10	9	1	2	1	3	48	28	60	R
9	4	10	6	4	3	2	3	68	32	52	7
0	3	10	9	1	4	4	1	60	40	52	8
A	3	8	7	3	4	2	3	84	34	76	H
B	2	6	5	4	1	2	1	32	20	60	8
C	4	10	9	1	4	5	1	72	68	32	O
D	3	9	6	4	4	4	2	56	56	40	O
E	3	8	4	4	1	-	3	52	48	32	C
F	3	9	6	3	4	3	1	76	56	24	P
G	3	9	7	3	2	1	4	40	24	52	6
H	4	9	10	-	4	1	4	68	48	44	N
I	3	8	6	4	2	1	2	52	16	56	L
J	3	8	5	5	4	1	4	60	40	48	I
K	3	7	6	2	4	-	5	52	12	76	R
L	4	10	10	-	4	4	1	92	76	20	K
M	4	9	9	-	3	2	1	84	36	48	N
N	3	8	6	3	4	1	3	44	16	52	M

Table <5.2.3.1> Results from Extension Testing (A)

Cat	# of Sub-cat.	Training Set			Known Set			Unknown Set			MOMC
		MED	RR		MED	RR		MED	RR		
		#cor	#cor	#rej	#cor	#cor	#rej	%cor	%cor	%rej	
O	3	10	8	2	5	5	-	72	40	44	D
P	4	10	9	1	4	1	3	80	40	48	F
Q	3	9	7	3	4	2	2	60	12	84	G
R	5	10	10	-	4	1	4	60	28	68	K
S	2	6	3	5	3	2	2	60	32	64	5
T	2	8	7	3	3	3	2	72	32	52	I
U	3	8	8	1	4	2	3	36	12	64	O
V	3	9	8	1	3	2	3	60	40	32	U
W	2	8	7	2	2	2	3	80	48	36	N
X	1	4	2	3	2	-	5	32	16	68	I
Y	3	10	9	1	4	4	1	56	44	20	X
Z	3	8	5	4	4	3	2	44	40	60	B
(	3	10	10	-	5	3	2	88	80	12	C
#	2	10	10	-	5	4	1	92	92	4	.
*	3	9	7	3	3	1	4	80	44	48	M
+	2	9	8	2	5	2	3	88	60	32	,
-	1	10	10	-	5	5	-	100	84	16	Nil
.	2	10	10	-	5	4	1	96	96	-	,
,	3	9	8	2	3	3	2	84	68	20	1
$	3	9	5	5	5	1	4	68	24	36	0
/	3	10	10	-	4	4	-	84	92	8	1
)	1	7	4	6	4	1	4	72	24	68	J
Tot	132										
% Total		86.7	72.1	23.0	70.8	44.3	48.3	65.5	41.9	45.7	

Legend: cor = correct rej = rejected
MOMC = most often mistaken category

Table <5.2.3.2> Results from Extensive Testing (B)

type of character set will be analyzed in succession (the learning set, first; and the known set, second; and the unknown set, last). The test results on the learning set show that the HPRS model did well on all but a few characters (namely X,4,B,S). The poor performance on these particular characters can be attributed to one or a combination of the following three conditions.

a) Because of the preprocessing method chosen, these characters are inherently difficult to categorize. That is, the PREP output for an X,4,B or S looks very similar to the output of some other characters (for example Y,A,E, or 5, respectively), thus making a large, well-defined category impossible.

b) Each of these categories suffered the loss of two subcategories during the final death process at the end of the training period. This accounts for two errors in each instance.

c) If near the end of the final training pass a new subcategory was formed which had a prototype point very close to the center of an existing subcategory, then the patterns belonging to the existing subcategory would be misclassified.

Conditions b and c could be eliminated if the training period was extended. Condition a is problem inherent in any pattern recognition scheme.

The overall performance of HPRS during the training period (86.7% correct in 6 passes) is superior to the Munson system using Condition 1 (65% correct in 18 passes), yet slightly inferior to his system under Condition 2 (84% correct in 4 passes). The Relative Reliability (RR) classification method is effective in rejecting many errors which resulted using the MED method. Only 4.9% of the characters were misclassified when RR classification was employed. Unfortunately a large number of correct patterns (14.6%) were rejected as well.

The test results on the known set show a sharp decline in machine performance over the results enjoyed by the training set. As was expected, a slight bias was present in the HPRS model towards correctly identifying characters from the known set as compared to characters from the unknown set. RR classification produced a large rejection rate - 48.3%; but, at the same time, only 7.4% of the known set were misclassified.

From the test results on the unknown set (65.5% correctly classified), it appears that the HPRS performance falls short of the performance of Munson's combined system. Some explanations can be offered to account for the poorer showing and these shall be discussed in Sec. 6.1, "Conclusions to the HPRS model". While HPRS does perform

better when classifying the known set (70.8% correctly classified), the results on the unknown sample (65.5% correctly classified) suggest that the ten alphabets chosen for training are quite "universal". Using the RR method of classification 45.7% of the unknown patterns were rejected while 12.4% were incorrectly identified. What is noticable throughout the entire results is that the characters which were poorly classified in the training set were poorly classified in the known and unknown sets as well. If the training performance in these few classes could be improved, the repercussions of such improvements would be far greater on the known and unknown character sets than on the training set itself.

In summary, this chapter has outlined the performance of the HPRS simulation model and the Munson model. An analysis of the HPRS performance has been presented along with a brief comparison with Munson's results when applicable. In the following and final chapter some conclusions shall be drawn about the overall research as described in this thesis.

CHAPTER VI

CONCLUSIONS AND OBSERVATIONS

Now that a presentation of the thesis research is completed, some conclusions and observations about the HPRS model and its applications to pattern recognition shall be stated. Sec. 6.1 contains the concluding remarks about the HPRS model as described in this thesis. Sec. 6.2 outlines some possible extensions to the model. In Sec. 6.3 the PREP-LEM recognition system is discussed in view of its applicability to pattern recognition - the character recognition problem excluded. The final section, Sec. 6.4, describes the contribution of the PREP-LEM system to the field of artificial intelligence. The title of this section is "Where to Go Next?"

6.1 Conclusion of the HPRS Model

As stated in the introduction of this thesis, "the immediate interest and purpose in this research effort is the exploration of a new, unique hand printing recognition system (HPRS)". The exploration has been completed. The model has been simulated in its entirety and has been tested extensively on over eighteen hundred hand-printed characters. The total performance of HPRS is slightly inferior to the Munson system. However, the range of the number of patterns

correctly identified for the individual classes easily encompasses the success rate achieved by Munson. "We are in the same ball park!" Some of the reasons for not achieving the excellent results reached by Munson are listed below.

1) The training set used in the HPRS experimentation consisted of only 460 characters. The training set employed by Munson numbered 4416 characters.

2) As mentioned in Chapter V, the poor performance in classifying a few particular characters, badly degraded the total performance of the HPRS model. If training were to continue until the classification performance of these characters reached an acceptable level (80%), the overall success of the system would be greatly improved.

3) While an attempt has been made to design a good, basic model, undoubtedly refinements could be made to HPRS to improve its performance. Some suggestions towards extending the present model are dealt with in the section following.

The main advantage of the HPRS model over the Munson system is its simplicity. The HPRS preprocessor, PREP, does not require the complicated timing gates and electrical circuitry needed by Munson's preprocessing system to extract features from sections of the character. For this reason it is highly probable that HPRS can process characters an order of magnitude faster than the Munson model. The great

reduction in hardware also makes HPRS less expensive to design and build.

HPRS transcends the Munson system because it possesses a Relative Reliability type of classification. If the rejected characters[*] could be sent to a special context analysis system to further aid in the classification, success rates comparable to human subjects might be achieved. However such context analyzers do not presently exist. More will be said about this subject in Sec. 6.4.

One major question has been left unanswered. Can HPRS be used as an effective recognition system on the basis of this simulation study? To fail to answer "Yes" or "No" might be considered "dodging" the question. However, on the basis of this study it is felt that further investigation is warranted, and that this investigation be completed with at least a simplified version of HPRS (i.e. using the PREP preprocessor but a somewhat modified classifier). Further

[*] Although no exact tally was made while collecting the results, it was noted that in almost all instances of rejection (over 90%) a correct subcategory appeared among the four most likely subcategories to which the unknown pattern might be assigned. In some cases three of the four subcategories chosen to represent the unknown character belonged to the correct category. Therefore, it seems reasonable that only those categories containing subcategories belonging to the set of four selected subcategories, need be passed to a context analyzer.

simulation might be useful but a warning should be posted at this point. It required over three and one half hours of CPU time on an IBM 360/67 to complete the extensive training and testing of the HPRS model. If further programming is undertaken, the interested researcher will have to be willing to pay the price.

To improve the basic HPRS model, however, some extensions will have to be made. The next section is devoted to a discussion of some possible changes and advancements.

6.2 Extensions to HPRS

It is the belief of many researchers in the field of character recognition (and more generally, the field of pattern recognition) that "edge" information is used in discriminating unlike characters. Indeed, there is some physiological evidence to suggest that sections of the visual cortex are devoted to edge detection to aid in the processing of visual information [Wooldridge (1963)]. In the optical transform the distance of any illumination from the center of the transform (the zero-order frequency for the x and y directions), is inversely proportional to the separation (or directly proportional to the spatial frequency) of the points of illumination occurring in the

light pattern of the object. Along any directional scan, the spatial separation between the edges of the printing is large relative to the spatial separation of points covered by the printing. Hence, when the diffraction pattern of an object is focused at the transform plane, the peripheral illumination contains information about the outline of the object*. In the simulation of PREP each point in the discrete Fourier transform is weighted equally. It is possible that if more weight is placed on values that are far from the transform center, better character discrimination could be achieved. This transform weighting idea is a certainly worth further investigation.

A second suggested extension to the HPRS model is the use of autocorrelation space rather than pattern space as the basis of classification [McLaughlin and Raviv (1968)]. The formula for the discrete case of the nth order autocorrection function of a pattern $X^{(t)}$ is given by

$$r_x(\tau_1,\tau_2,\ldots,\tau n) = \sum_t x(t)x(t+\tau_1)\ldots x(t+\tau n) \quad ..(6.2.1)$$

---

*In optical filtering techniques (described by Dobrin [1968]) an opague disk is centered along the xy axis of the transform plane so that only high spatial frequencies are passed (high pass filter). If characters are reconstructed after high pass filtering, they contain bright edges. This transform characteristic justifies the weighting of peripheral transform information so that character edges may be emphasized.

As an example, the vector X=<1 3 2 1> would have the autocorrelation values of r_x=<7 12 17 6>. The important property of an autocorrelation function is that it is invariant to translation. That is, if y=<3 2 1 1> then y=<7 12 17 6>. This invariance property is particularly interesting in view of the surface fitting problem. Typically the output from PREP is in matrix notation which is representative of the combined preprocessed surface. To perform an nth-order autocorrelation over the matrix, M, an autocorrelation (as described in Eq.(6.2.1)) should be taken first along the rows and then along the columns of M*. This procedure can be described mathematically as

$$r_M(\tau_1,\tau_2,\ldots,\tau_n) = \sum_{1} r(t,k)\cdot r(t,k+\tau_1)\cdot r(t,k+\tau_2) \ldots r(t,k+\tau_n) \qquad \ldots (6.2.2)$$

where $r(t,k)=r_{M_k}(\tau_1,\tau_2,\ldots,\tau_n)=\sum_{t} M(t,k)\cdot M(t+\tau_{}\,,k) \ldots$

$$M(t+\tau_n,k) \qquad \ldots (6.2.3)$$

[*] The nth order autocorrelation over M may be calculated along the columns first and then across the rows. While the result will be different from the row-column procedure, both methods are invariant to translation. Either method maybe applied successfully as long as one is used consistently throughout.

McLaughlin and Raviv [1968] have shown that autocorrelation techniques have reduced classification errors up to 100% over noncorrelation methods. Nowhere in the literature could information be found on autocorrelation functions for the three variables (surface fitting) case.

While autocorrelation functions appear very attractive they have a major drawback. Because of the multitude of calculations that are required, a hardware implementation of a two-dimensional (NxN) autocorrelator is extremely complicated.* However, the development of a fast, hardwired autocorrelator would likely better the performance of HPRS. If added to the HPRS model, the autocorrelator is situated between the preprocessor, PREP, and the categorizer, LEM. By using this design, patterns are classified on the basis of autocorrelation space rather than pattern space.

Because the theory that is developed for the categorizer is substantually heuristic, any extensions or changes to LEM would by application dependent. That is, whatever classifying method produces the best results is the acceptable one. The following section discusses some other applications of the PREP-LEM recognition system in the general field of pattern recognition.

---

*No information could be found on the design of a two-dimensional autocorrelator, but with reference to Eq.(6.2.2) and Eq.(6.2.3) the architecture would appear to be very complex.

6.3 Applications

While the HPRS model is tested for one application-character recognition-a special effort has been made to insure that the model components, the PREP preprocessor and the LEM categorizer, are not application dependent. This section will be confined to a discussion of the recognition system (composed of PREP and LEM) as applied to other pattern recognition problems. It should be made clear, however, that over-generalization can also be harmful. The PREP-LEM system has not been tested for any of the applications described below-applications which are far more complex than the character recognition problem.

One possible application of the PREP-LEM system is in the area of oil prospecting. Pictorial representations of reflection data of the earth's surface are collected in bulk by seismic operators. Most of the data is "negative", however; hundreds of pictures must be scanned so that the few "positive" cases may be extracted for closer examination. Using the pattern recognition system developed in this research, large scale examinations could be completed in seconds. The frequency analysis as performed by the preprocessor, PREP, is especially suited to the reflection data which is composed of hundreds of seismic waves. To enable the categorizer to eliminate negative instances, LEM need only be presented with a learning set which adequately

describes the situations in which oil is recoverable.

A second application in which the PREP-LEM system might be used is in fingerprint analysis. The spatial frequencies of fingerprint data could be characterized by the optical transform technique of the PREP preprocessor. To search through all the recorded fingerprints for a set identical to the unknown set is a tremendous task. Instead LEM could be employed to aid in categorizing some of the recorded sets of fingerprints. From the categorization a file could be made for each category that is formed. When a set of fingerprints requires identification, they are classified by LEM which points the examiner to the proper file. The examiner may then choose to scrutinize the file himself or allow LEM to pick out the most closely matched set. This procedure is undoubtedly an oversimplification of the difficulties involved in fingerprint identification. Any recognition problem which considers millions of unique patterns requires a multitude of refinements before an acceptable solution is reached.

A third application (the final application to be discussed in detail) is in the field of medicine. While there are a large number of pattern recognition applications in medicine, only the detection of Regional Myocardial (heart muscles) Blood Flow activity is outlined here. To measure the blood flow through the heart muscles, a radioactive

distribution is created by injecting the Xe^{133} isotope into the coronary artery. The gamma ray emission which is measured by a gamma ray camera is directly proportional to the blood flow intensity. The gamma ray camera scans (in a matrix-like fashion) the entire heart recording the amount of radiation at each of the nxn detection points. The detection matrix is feed into the PREP where it is preprocessed. The preprocessor data (an nxn matrix) is passed to the LEM categorizer where a surface fit is made between the preprocessor data and the stored prototype matrices. If the intensity of radiation is low in certain areas of the heart (indicating poor blood flow) then LEM should be designed to detect the corresponding pictures and record them for further scrutiny by a human observer.

The PREP-LEM pattern recognition system might also be employed in the areas of weather prediction from aerial photographs, radiation detection in nuclear reactors, and underwater scanning in the field of oceanography. Now that the model has been summarized, extensions have been suggested and further applications have been outlined, a final question remains unanswered...

6.4 Where to Go Next?

The content of this thesis is embodied in a survey

of the field of character recognition, a summary of the theory developed for the HPRS model, an outline of a hardware schemata and simulation of HPRS, and a tabulation and discussion of the experimental results. Until this section, the presentation has not contained the usual exposition on the relevance of character recognition (or pattern recognition more generally) to the study of artificial intelligence. The author has planned this action purposely, for it is his contention that neither this, or any other pattern recognition device to date, displays intelligence. However a discussion of this nature does fall into this section, "Where to go next?", since the field of pattern recognition should be directed towards the production of some form of machine intelligence. It must be agreed that recognition machines are now capable of operating at comparable speeds to humans and they certainly do not suffer from fatigue. But machines do not enjoy the recognition accuracy that the human observer does. Because man possesses intelligence, he is capable of identifying highly distorted patterns through context analysis.

Two recognition systems which possess contextually aided recognition capabilities were designed by Duda and Hart [1968] and Bledsoe and Browning [1966]. Both of these systems were successful within restricted vocabularies. Duda and Hart developed a program which identifies hand-

printed characters from FORTRAN programs written on coding sheets (characters that were similar to Munson's data). They received excellent results (reducing the error rate from 9.3 percent without context analysis to 2.4 percent with context analysis) by applying the syntactical and semantic rules of the FORTRAN II language to their context analyzer. The Bledsoe and Browning model correctly identified up to 94% of the letters using context-positioning with a limited vocabulary of 677 words.

While the above systems only considered characters from special or limited vocabularies, they do point in the direction of the next step to be taken in the field of pattern recognition. Recognition machines must be capable of context analysis before they can ever become accurate and commercially useful. Indeed, this is "Where to go next?"

BIBLIOGRAPHY

Andrews, H.C. and Pratt, W.K., 1968. "Digital Computer Simulation of Coherent Optical Processing Operations", Computer Group News, IEEE, pp. 12-19, Nov. 1968.

Barkdoll, I.H., McGlamery, B.L., 1968. "An On-line Image Processing System", Proceedings - 1968 ACM National Conference, Brandon/Systems Press Inc., Princeton, N.J., pp. 705-716.

Ball, G.H., Hall, D.J., 1964. "Some Fundamental Concepts and Synthesis Procedures for Pattern Recognition Preprocessors", paper presented at the International Conference of Microwaves, Circuit Theory, and Information Theory, September 7-11, 1964, Tokyo, Japan.

Bledsoe, W.W. and Browning I., 1966. Pattern Recognition, L. Uhr, edt., John Wiley & Sons, Inc., New York, pp. 301-316.

Dickenson, A. and Watrasiewicz, B.M., 1968. "Optical Filtering Applied to Postal Code Reading," I.E.E. N.P.L. Conference on Pattern Recognition, C. Baldwin Ltd., Tunbridge Wells, Kent, pp. 207-219.

Dobrin, M.B., 1968. "Optical Processing in the Earth Sciences," IEEE Spectrum, pp. 59-66, September 1968.

Duda, R.O. and Fossum, H., 1966. "Pattern Classification by Iteratively Determined Linear and Piecewise Linear Discriminant Functions," IEEE Trans. on Electronic Computers, vol. EC-15, pp. 220-232, April 1966.

Duda, R.O. and Hart, P.E., 1968. "Experiments in the Recognition of Hand-Printed Text: Part II--Context Analysis", 1968 Fall Joint Computer Conference, May 1968.

G-AE Subcommittee on Measurement Concepts, 1967. "What is the F.F.T.?" IEEE Transactions on Audio and Electroacoustics, vol. AU-15, p. 45, June 1967.

Harrison, Michael A., 1965. Introduction to Switching and Automata Theory, McGraw-Hill Co., Toronto.

Ho, Y.C. and Agrawala, A.K., 1968. "Pattern Classification Algorithms," Proceeding of IEEE, p. 2100, Dec. 1968.

Holeman, J.M., 1968. "Holographic Character Readers", Pattern Recognition, Thompson Book Co., Wash., D.C., pp. 63-78.

Holt, A.W., 1968. "Comparative Religion in Character Recognition Machines", Computer Group News, IEEE, pp. 4-11, Nov. 1968.

Hunt, E.B., Marin J., Stone, P.J., 1966. Experiments in Induction, Academic Press, New York.

Kazmierczak, H. and Steinbuck, K., 1963. "Adaptive Systems in Pattern Recognition", IEEE Transactions of Electronic Computers, pp. 822-835, Dec. 1963.

Korpel, A. 1968. "Acoustic Imaging and Holography", IEEE Spectrum, vol. 5, Number 10, pp. 45-52, October 1968.

Lesen, L.B. and Hirsch, P.M., 1968. "Computer Synthesis of Holograms for 3-D Display", Communications of the ACM, pp. 661-674, Oct. 1968.

Lugt, A.V., 1968. "A Review of Optical Data-Processing Techniques", Optical Acta, vol. 15, no. 1, pp. 1-13.

McLaughlin, J.A., and Raviv, J., 1968. "Nth-Order Autocorrelations in Pattern Recognition", Information and Control, vol. 12, number 2, February 1968.

Minsky, M., 1963. "Steps Toward Artificial Intelligence", Computers and Thought, McGraw-Hill, Toronto, pp. 406-450.

Nagy, G. 1968. "State of the Art in Pattern Recognition", Proceedings of IEEE, p. 836, May 1968.

Nilsson, Nils J., 1965. Learning Machines, McGraw-Hill Book Company, Toronto.

Parzen, E., 1967. "Informal Comments on the Uses of Power Spectrum Analysis", IEEE Transactions on Audio and Electroacoustics, vol. AU-15, p. 74, June 1967.

Rabinow, J. 1968. "The Present State of the Art of Reading Machines", Pattern Recognition, Thompson Book Co., Wash., D.C., pp. 3-27.

Rosen, C.A. and Hall, D.J., 1966. "A Pattern Recognition Experiment with Near-Optimum Results", IEEE Trans. on Electronic Computers, vol. EC-15, pp. 666-667. August, 1966.

Saenger, E.L., 1966. "The Use of Computers in Nuclear Medicine", Proc. of Conf. on the Use of Computers in Radiology, October 1966.

Sande, G., 1968. "A Fast Fourier Transform Subroutine", University of Alberta Department of Computing Science Program Library, May, 1968.

Sebestyen, G.S., 1962. Decision-Making Processes in Pattern Recognition, MacMillan Co., New York.

Selfridge, O.G. and Neisser, U., 1963. "Pattern Recognition by Machine", Computers and Thought, McGraw-Hill, Toronto, pp. 237-250.

Smith, F.D., 1968. "How Images Are Formed", Scientific American, pp. 96-108, September 1968.

Stark, L., Okajima, M. and Whipple, G.H., 1962. "Computer Pattern Recognition Techniques: Electrocardiagraphic Diagnosis," Comm. of the A.C.M., vol. 5, pp. 527-532, October, 1962.

Steinbuck, K., 1965. "Adaptive Networks Using Learning Matrices", Kybernetik, pp. 148-152, Feb. 1965.

Steinbuck, K. and Piske, U.A.W., 1963. "Learning Matrices and Their Applications", IEEE Transactions on Electronic Computers, pp. 846-862, Dec. 1963.

Syms, G.H., 1967. "A Pattern Recognition Model For On-line Curve Fitting: An Application of Threshold Theory", (Doctoral Thesis), University of Washington, August 1967.

Uhr, L. and Vossler, C., 1963. "A Pattern-Recognition Program That Generates, Evaluates and Adjusts its Own Operators", Computers and Thought, McGraw-Hill, Toronto, pp. 251-268.

Wooldridge, D.E., 1963. The Machinery of the Brain, McGraw-Hill, Toronto.

APPENDIX A

THE DERIVATION OF THE ONE-STEP ERROR CORRECTION β FACTOR

The general error correction formula is given by

$$P_j'^{(i)} = \beta P_j^{(i)} + (1-\beta)X \qquad \ldots \quad (A1)$$

In one-step error correction the prototype point $P_j^{(i)}$ is moved to within a designated distance of the input pattern vector X. In Eq.(3.2.4.3) the designated distance is the threshold value $T_j^{(i)}$. Suppose $P_k^{(m)}$ is the prototype point of the nearest neighbor subcategory to the subcategory $C_j^{(i)}$. Then to find the proper $P_j'^{(i)}$ (such that it falls within $T_j^{(i)}$ of X), the β factor from Eq.(A1) must be calculated.

It is known that

$$g_j^{(i)}(X)<T_j^{(i)}$$

or else no adjustment would be necessary,

$$g_j^{(i)}(X) = P_j^{(i)}X-1/2P_j^{(i)2}-1/2X^2 \qquad \ldots \quad (A2)$$

To adjust $P_j^{(i)}$ such that $g_j^{(i)}(X) \geq T_j^{(i)}$, $P_j'^{(i)}$ from Eq.(A1) is substituted into Eq.(A2). Therefore

$$T_j^{(i)} \leq \beta^2 (P_j^{(i)} \cdot X-1/2P_j^{(i)2}-1/2X^2)$$

By substituting Eq.(A2), the result is

$$\beta^2 \leq \frac{T_j^{(i)}}{g_j^{(i)}(X)}$$ (the inequality is switched because $g_j^{(i)}(X)<0$)

Therefore $$\beta \leq \left[\frac{T_j^{(i)}}{g_j^{(i)}(X)} \right]^{1/2}$$